# UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y SISTEMAS DE TELECOMUNICACIÓN**


**TRABAJO FIN DE MÁSTER**

**MÁSTER UNIVERSITARIO EN INGENIERÍA ACÚSTICA DE LA EDIFICACIÓN Y MEDIO AMBIENTE**

# ESTUDIO DE VERIFICACIÓN BIOMÉTRICA DE VOZ


**MARTA AGUIRREZABALA AGUSTÍN**


JULIO 2015

*A mis padres*

# Agradecimientos

En primer lugar me gustaría agradecer a Borja Lacabex, su alegría, su saber estar y su intento de contagiar esa calma que a mi me falta. Y sobre todo su paciencia.

A mis padres por predicar con el ejemplo de eso que dicen que el único sitio en el que el éxito llega antes que el trabajo es en el diccionario. A mi madre por esos tuppers que te dan fuerzas para hacer siete proyectos más. A mis hermanos Iñigo y Aitor, por hacer que merezcan la pena las horas invertidas en ese autobús hasta casa y que ir al circo resulte hasta aburrido.

Y por supuesto, a mis abuelos por su ilusión y por demostrarme que mi felicidad es también la suya.

**GRACIAS**

# Abstract

En este proyecto estudia la posibilidad de realizar una verificación de locutor por medio de la biometría de voz. En primer lugar se obtendrán las características principales de la voz, que serán los coeficientes MFCC, partiendo de una base de datos de diferentes locutores con 10 muestras por cada locutor.

Con estos resultados se procederá a la creación de los clasificadores con los que luego testearemos y haremos la verificación. Como resultado final obtendremos un sistema capaz de identificar si el locutor es el que buscamos o no. Para la verificación se utilizan clasificadores Support Vector Machine (SVM), especializado en resolver problemas biclase.

Los resultados demuestran que el sistema es capaz de verificar que un locutor es quien dice ser comparándolo con el resto de locutores disponibles en la base de datos.

# Abstract

Verification based on voice features is an important task for a wide variety of applications concerning biometric verification systems. In this work, we propose a human verification though the use of their voice features focused on supervised training classification algorithms. To this aim we have developed a voice feature extraction system based on MFCC features.

For classification purposed we have focused our work in using a Support Vector Machine classificator due to it's optimization for biclass problems. We test our system in a dataset composed of various individuals of different gender to evaluate our system's performance. Experimental results reveal that the proposed system is capable of verificating one individual against the rest of the dataset.

# Índice general

# Índice de tablas

# Índice de figuras

# Capítulo 1

# Introducción

## 1.1. Motivación

El primer estudio Biométrico fue desarrollado en el año 1880 por Alphonse Bertillon. Éste, realizó los primeros estudios antropométricos mediante un sistema de caracterización de individuos con el fin de identificar criminales [2].


Figura 1.1: Biometría utilizada en el año 1880

Este método, tal y como se ve en la figura 1.1, consiste en tomar medidas de diferentes partes del cuerpo de los criminales, con el objetivo de tenerlos controlados [3]. Dicho procedimiento se utilizó hasta que apareció la técnica biométrica de identificación mediante las huellas dactilares. Fue más adelante cuando se comenzó a estudiar la posibilidad de utilizar el iris como rasgo identificativo y ya es en 1965 cuando se empieza a plantear el

uso del habla como rasgo característico para identificación y/o verificación de humanos [4].

Tal y como hacía Bertillon, para poder realizar este tipo de sistemas, es necesario estar dado de alta en ellos, el sistema debe conocer al usuario a verificar o identificar. De esta forma se crean plantillas (tanto al darse de alta como en cada proceso de autenticación o verificación). Los datos se tratan de diferentes maneras, con el objetivo de obtener las características particulares para cada individuo. Dependiendo de cual sea el rasgo característico que se esté analizando se obtendrán un tipo de características u otras.

En este tipo de sistemas, la seguridad y lo que minimiza la tasa de error es el rasgo o la característica que se toma, la precisión o la calidad con la que se captura y el umbral de aceptación que fijemos. Por ello, hay que elegir cuidadosamente tanto la característica, como el umbral en el proceso de verificación. Con estos parámetros se ajusta el grado de fuerza y la calidad del sistema.

Hasta hace muy poco, el reconocimiento por huella dactilar era prácticamente el único método biométrico fiable utilizado [5]. Sin embargo, a medida que avanzaba la tecnología y la aplicación de la misma en entornos tanto de ocio o comodidad como de seguridad, la evolución en desarrollo biométrico es muy grande. Hoy en día hay muchos sistemas comerciales de uso generalizado que utilizan la biometría.

Además, la demanda de la seguridad tanto por parte de empresas como de usuarios ha hecho que proyectos de verificación biométrica ya sea por voz, por huella dactilar etcétera estén en auge. Como muestra de ello grandes empresas como Nuance y 3M se han decantado por investigar y diseñar sistemas que aporten al usuario seguridad basándose en la biometría.

Los principales ámbitos de aplicación en los que se trabaja la biometría son: la salud y la seguridad, para verificación de firmas por ejemplo. El más relevante y en el que más tiempo y dinero se invierte es en seguridad. Hasta ahora estos sistemas se pueden separar en dos grandes módulos [1]:

- Algo que el usuario sabe (contraseña).
- Algo que el usuario tiene (tarjeta personal).

Las diferencias frente a un sistema de seguridad tradicional es que con la biometría se va un paso más allá, creando un nuevo módulo en los sistemas de seguridad. Este módulo analizara algo que el usuario es o hace.

Este concepto se puede dividir en dos secciones, tal y como aparece en la figura 1.2:

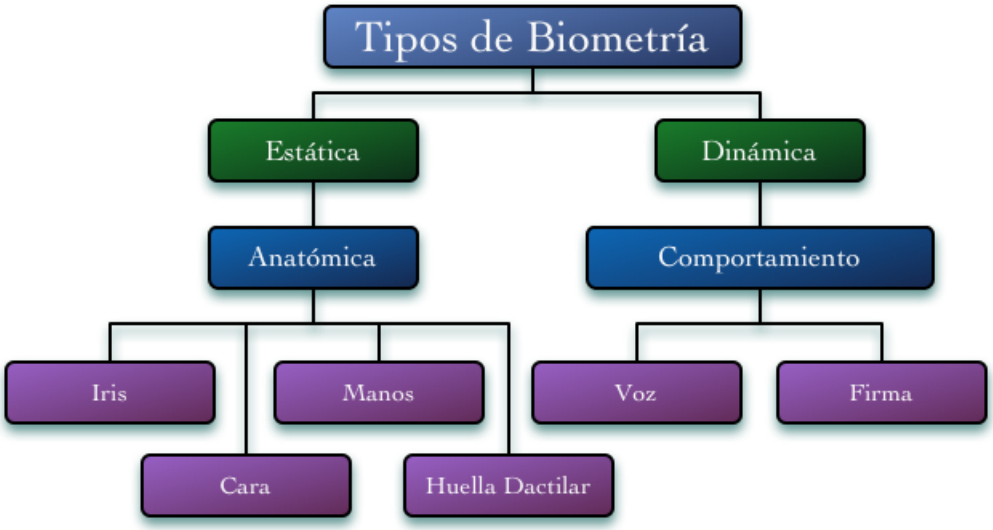


Figura 1.2: Tipos de Biometrías. (UNAM - Facultad de Ingeniería Biometría Informática).

Por un lado se reconoce la Biometría estática, la cual define algo que el usuario es, un rasgo físico o anatómico, ya sea la huella dactilar, la cara, las líneas de la mano... que es un rasgo característico y único en cada ser humano. Y por otro lado se encuentra la Biometría dinámica la cual habla de la conducta o del comportamiento, algo que el humano hace, como su escritura, sus gestos, su forma de caminar, movimientos corporales o su propia voz, que es el objeto de estudio de este proyecto.

La Biometría permite realizar estudios de reconocimiento de humanos basados en rasgos conductuales o físicos intrínsecos y particulares de cada ser humano. La Biometría permite autenticar individuos.

Además existen dos opciones de autenticación; la identificación y la verificación. La identificación dice quién es una persona dependiendo de sus características físicas o de su conducta. En contra, la verificación biométrica aclara si una persona es quien dice ser, partiendo de análisis biométricos y realizando comparaciones con otros posibles candidatos.

Este estudio se centra en la verificación de usuario mediante la voz. La Biometría de Voz es muy utilizada en aplicaciones o sistemas relacionados con la seguridad. Ya que cada individuo tiene unas características físicas diferentes.

La voz se considera una característica física, un identificativo como podría ser la huella dactilar. Se considera característica única y diferencial con respecto a otros humanos debi-

do a que la voz surge del tracto vocal, el cuál contiene cavidades de dimensiones variables y singulares para cada individuo.

La Biometría de Voz utiliza el pitch, el tono, el ritmo del habla... Aunque no es tan sencillo realizar este tipo de software ya que pueden existir inconvenientes tales como el ruido de fondo, enfermedades, edad, estados de ánimo en los cuales pueden cambiar la voz y también pueden existir diferencias entre sistemas de captación de la señal, ajenos al locutor. Sin contar con que el usuario puede tratar de estafar al sistema realizando imitaciones, pero este tema se deja de lado en esta investigación inicial.

Para el desarrollo de este proyecto se realiza una extracción de características del usuario a verificar y se procede a comparar el individuo a autenticar con otros candidatos. Previamente el individuo declara su identidad y el objetivo es averiguar si es quien dice ser. Para ello deben compararse las características biométricas extraídas del individuo con las almacenadas para ese individuo y otros en la base de datos, y decidir si concuerdan lo suficiente. Si no es así, se deniega el acceso.

En cuanto a Biometrías de Voz existen diferentes tipos:

- Dependiente de texto.
- Independiente del texto.

El principal tipo de aplicación de la interacción por voz son los conocidos como Sistemas de Diálogo Hablado (dependientes de texto). A la hora de implementarlos, principalmente se pueden optar por dos tipos de estrategias. La primera de ellas consiste en el uso de un lenguaje basado en comandos o palabras aisladas. Así, el usuario podría utilizar un comando como usar las palabras “Sí” o “No” en una diálogo dirigido, o utilizar como comandos una serie de números para seleccionar una opción en un sistema de asistencia telefónica en vez del sistema DTMF (Dual Tone Multiple Frequency) tradicional [15]. La segunda opción es utilizar el lenguaje natural, o uno restringido a ciertas estructuras gramaticales.

Por otro lado, la Biometría independiente del texto resulta atractiva debido a que se puede aplicar a sistemas de seguridad, realmente hay ocasiones en las que se necesita mayor seguridad para ciertas actividades o simplemente es que se nos puede olvidar una clave o puede ser sustraída. Por eso se ha ido desarrollando proyectos biométricos, además

de por la comodidad que le aporta al usuario, ya que no es necesario que recuerde una contraseña o que lleve una tarjeta o una identificación consigo.

La voz es la forma más natural de comunicación entre humanos. En este sentido, el uso del procesamiento de la voz como un modo de interacción con los sistemas o de acceso a ellos, ofrece innumerables ventajas respecto a los interfaces clásicos de comunicación con los sistemas.

- Bajo costo.
- Fácil uso y aceptado por los usuarios.
- Concepto natural (cuando nos llaman por teléfono. El primer instinto es intentar reconocer a la persona).
- La voz se puede capturar y transmitir de una manera simple a través de dispositivos cotidianos como el teléfono fijo o móvil.
- La única biometría que te permite identificarte de forma remota.
- Aplicaciones de poco tamaño que pueden ser almacenadas en tarjetas SD, teléfonos, FPGAs, entre otras.
- Mientras que otros interfaces requieren que el usuario centre completamente su atención en esa actividad, un interfaz vocal libera al usuario de la necesidad de usar sus manos y la vista para poder realizar otras tareas simultáneas a la interacción.

Sin embargo, este tipo de sistemas presenta una serie de inconvenientes y dificultades que pueden hacer no apropiada su utilización:

- No es la biometría más segura.
- La voz humana cambia constantemente (juventud, edad adulta, estados de ánimo, enfermedad).
- Spoofing.
- Susceptibilidad al canal de transmisión y las variaciones del micrófono y su ruido.
    - Ruido de fondo.
    - Falta de cobertura si se utiliza de forma remota.

- Problemas del usuario para hablar.
- Se necesita un modelo de voz claro y limpio, esto es, una buena referencia.

La Biometría de Voz resulta muy interesante debido a que la señal la cual queremos analizar es muy fácil de capturar y su grabación no requiere demasiada molestia para el usuario. Aunque por otro lado, hay que tener en cuenta que la señal se puede degradar fácilmente tanto voluntaria como involuntariamente. El usuario puede cambiar su tono de voz si quiere o puede verse degradada por ruido de fondo.

Actualmente la biometría sigue sin tener un rendimiento perfecto, tienen una tasa de acierto que se mueve en un rango desde el 60 % hasta el 99,9 % de acierto.

Para conocer este dato hay que analizar los resultados obtenidos en el proceso de verificación. Analizaremos el número de aciertos frente al número de observaciones realizadas, para un conjunto de muestras positivas y negativas.

En un problema de clasificación binaria, para la verificación cada usuario dispone de una clase positiva y otra negativa, encontramos cuatro tipo de situaciones en cuanto al resultado obtenido: el número de Verdaderos Positivos (TP), el de Falsos Positivos (FP), Falsos Negativos (FN) y Verdaderos Negativos (TF). A partir de estos cuatro datos podemos construir la Matriz de Confusión, tal y como mostramos en la figura 1.3, que nos permitirá elaborar las diferentes gráficas para la evaluación del sistema desarrollado.

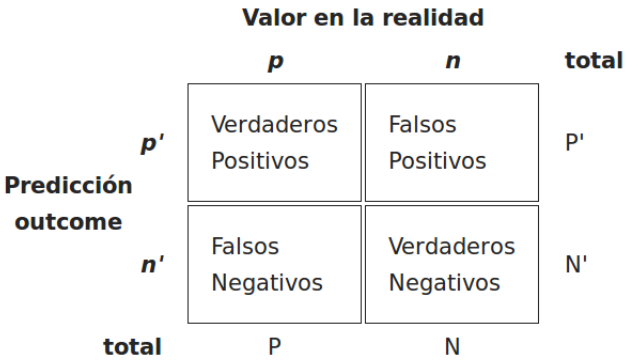


Figura 1.3: Matriz de confusión.

Si el resultado de la clasificación corresponde con el deseado, en caso de que sea positivo hablaremos de Verdadero Positivo y Falso Positivo en caso negativo. Si por el contrario

el resultado esperado no corresponde con el deseado, en caso de que sea positivo el obtenido y negativo el esperado hablaremos de Falso Positivo. Lo mismo en caso de que el obtenido sea negativo y el esperado sea positivo, en este caso se hablará de Falso Negativo.

La curva más representativa para llevar a cabo la evaluación de cualquier clasificador es la curva ROC. Esta curva se basa en dos parametros: $sensivity = TP/TP + FN$ para el eje $y$, $1 - specifity = TN/TN + FP$ para el eje $x$. Cuanto más cerca se encuentre la curva del punto $(x, y) = (0, 1)$ el sistema desarrollado será óptimo para llevar a cabo el proceso de clasificación. En la figura 4.28 observamos diferentes tipos de curvas ROC con resultados más o menos óptimos.

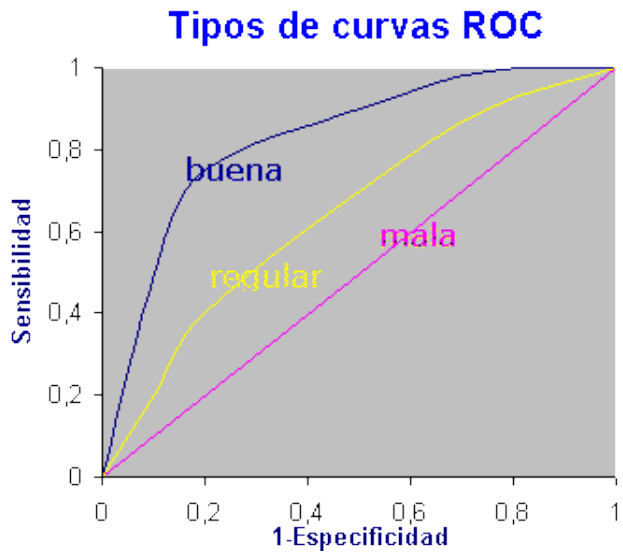


Figura 1.4: Ejemplos de curvas ROC y su resultado

## 1.2. Objetivos y alcance

### 1.2.1. Objetivos

Como ya hemos comentado anteriormente, el objetivo de este proyecto es evaluar y analizar diferentes voces humanas con el fin de verificar si el locutor es quien dice ser. Se pretende analizar resultados obteniendo previamente muestras de diferentes voluntarios con el fin de tener tanto muestras positivas como negativas de los usuarios. Este objetivo principal, lo podemos describir en varios requisitos más concretos.

- Análisis de señales de voz.

- Obtención de bases de datos de voz.
- Análisis y extracción de características humanas de la voz.
- Comprobación de la existencia de signos distintivos únicos para cada una de las características obtenidas.
- Evaluar las posibilidades de la biometría de voz utilizando software libre.
- Analizar las posibilidades de verificación con sistemas sencillos de grabación (como micrófono de móvil o de ordenador).
- Estudio de diferentes modelos de clasificadores para la verificación final.
- Creacion de entrenamientos de los clasificadores.
- Testeo de los clasificadores ya entrenados.
- Integrar las aplicaciones software realizadas en una única aplicación (PDS + SVM).

### 1.2.2. Alcance

En el alcance definiremos las tareas que vamos a realizar en el proceso de este proyecto para poder cumplir con los objetivos propuestos anteriormente.

- Estudiar alternativas de software libre como puede ser Python.
- Elegir y definir la herramienta software para que sea lo más efectiva posible a la hora de desarrollar y visualizar.
- Estudiar técnicas de procesado de señal de audio para posteriormente trabajar con ellas fácilmente.
- Estudiar diferentes características de la voz.
- Programación y obtención de datos tangibles de esas características.
- Análisis de la exclusividad que ofrecen esas características.
- Estudio de diferentes sistemas de clasificación.
- Eleccion de sistema de clasificación para la verificación de usuario.

- Entrenamiento de clasificadores con las características elegidas.
- Testeo de clasificadores con las características elegidas.
- Realizar pruebas analizándolas para obtener conclusiones y poder realizar mejoras.

## 1.3. Justificación

En esta sección se explica brevemente de donde surge la motivación para la realización de este estudio. Por un lado se explica la motivación social y por otro se justifica la motivación económica.

### 1.3.1. Justificación Social

La biometría ha formado parte de nuestras vidas desde hace más de un siglo y medio, aportando seguridad a la sociedad. Se ha tratado de estudiar tanto el comportamiento humano como sus rasgos físicos más característicos para destacar las particularidades de cada individuo y así tenerlos identificados.

Lo principal y lo que más destacamos en este proyecto es la posibilidad de verificar a alguien por su voz, no solo por temas de seguridad si no también por comodidad para el usuario. Ya que crear un identificativo para la voz significa no obligar al usuario a llevar una tarjeta identificativa consigo mismo o tener que recordar una contraseña.

Todos los tipos de biometría funcionan de la misma manera, se parte de una plantilla creada con características del usuario se compara y dependiendo de un umbral se acepta o no se acepta. Este umbral será el que aporte mayor o menor seguridad, todo depende de lo permisivos que seamos con nuestro sistema y de cuantas muestras tengamos en nuestra plantilla para cada usuario.

A pesar de que la biometría de voz está teniendo una buena aceptación por parte de la sociedad, hay otro tipo de biometrías que aún sacan mucha ventaja a esta, ya que al llevar más tiempo en el mercado, la sociedad está más familiarizada con ellas.

En la figura 1.5 vemos un ejemplo de como está actualmente la biometría de voz frente a otro tipo de biometrías [16].

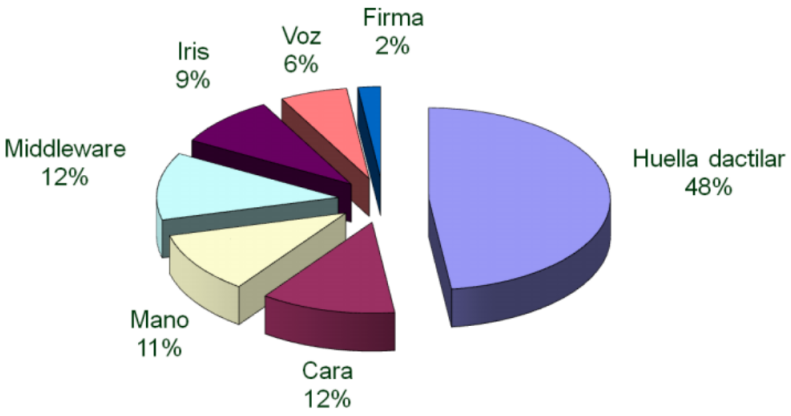


Figura 1.5: Cuota de mercado en tecnología biométrica.

### 1.3.2. Justificación Económica

En la justificación económica explica por qué resulta mejor este producto frente a otras Biometrías disponibles en el mercado.

El motivo principal es que este tipo de biometría no necesita nada que la mayoría de los usuarios no tengan. Para poder utilizar este tipo de sistemas simplemente es necesario algo que permita realizar capturas de señales de voz, como un micrófono, ya sea en el móvil, en la Tablet o en un ordenador. Esto hace que para un uso futuro no se requiera que el usuario se gaste el dinero en complejos sistemas de análisis biométricos como un lector de huella dactilar o de iris.
Por otro lado, la tecnología con la que se ha desarrollado el sistema no solo ha sido escogida por dar buenos resultado y cumplir con las funciones necesarias para el correcto desarrollo del proyecto, sino que también se ha mirado el precio en relación con la calidad. Se ha tomado como requisito la utilización de software libre debido a los aportes realizados por la comunidad de software libre y código abierto el cual permite entender más fácilmente y desarrollar de forma más rápida el proyecto.

## 1.4. Fases

Para el desarrollo de este proyecto hemos seguido las fases que podemos ver en la figura 1.6. A partir de este pequeño esquema explicaremos cómo se ha ido desarrollando el proyecto.


Figura 1.6: Diagrama de fases de desarrollo del proyecto.

Como paso inicial, se ha detectado el problema y se ha acotado con el objetivo de conocer que puntos queremos tratar y qué puntos no. Además se realizó una demo en Matlab para corroborar que el proyecto era factible y que la programación a pesar de no ser sencilla iba a ser posible en un límite de tiempo establecido y utilizando parte de los conocimientos adquiridos en el Máster.

El segundo paso es el nombrado en la figura como Desarrollo. En este punto se ha procedido a buscar información acerca de otros sistemas de Biometría de Voz y conocer qué procedimientos se han seguido en ellos para realizar un estudio del mercado actual y coger las ideas que más se adapten a lo que buscamos. Además se necesitan conocimientos en el ámbito del procesado de señal por lo que fue necesario un estudio a fondo para poder trabajar cómodamente con las señales. Fue de gran ayuda el curso Audio Signal Processing for Music Applications impartido por Xavier Serra de la mano de la Universidad de Standford[1]. Gracias a este curso hemos adquirido conocimientos en cuanto a características particulares de la señal de voz, y refrescado lo aprendido en cuanto a tratamiento y manipulación de señales acústicas.

El siguiente paso se conoce como Pruebas de campo, en el cual se implementa lo dicho anteriormente y se realizan diferentes pruebas con el objetivo de obtener conclusiones y

[1]http://www.dtic.upf.edu/ xserra/

acercarse hacia el último paso nombrado en la figura como Mejora. En esta sección se analizan los resultados obtenidos. Además, se deberá volver de nuevo al segundo punto en caso de que los resultados no sean los deseados. Para ello se volverán a estudiar nuevas características, se implementaran y se testeará, obteniendo nuevas conclusiones.

## 1.5. Planificación

El diagrama de Gantt es una herramienta de planificación permite ver la distribución temporal de tareas y recursos, como el que podemos ver en la figura 1.7. En este diagrama se explica cuales han sido las etapas a tener en cuenta y la duración estimada para cada una de las tareas establecidas para la finalización del proyecto. En las columnas tendremos el eje temporal con la duración de cada una de las tareas y en las filas tendremos colocadas cada una ellas.

Además vemos el diagrama temporal donde gráficamente está la duración de cada tarea a realizar y si se superponen o no.

Además de esto, para una planificación más especifica, encontraremos un desglose de tareas dividiendo cada tarea en diferentes tablas y especificando cual es la descripción de cada tarea, el objetivo que se pretende cumplir, si estas tareas tienen subtareas, cuales son los resultados esperados, las fechas propuestas y quien es el responsable de que dicha tarea se realice bien y en el tiempo previsto. Ya que si alguna de las tareas se superpone a otra puede ocasionarnos problemas en cuanto al tiempo de finalización.

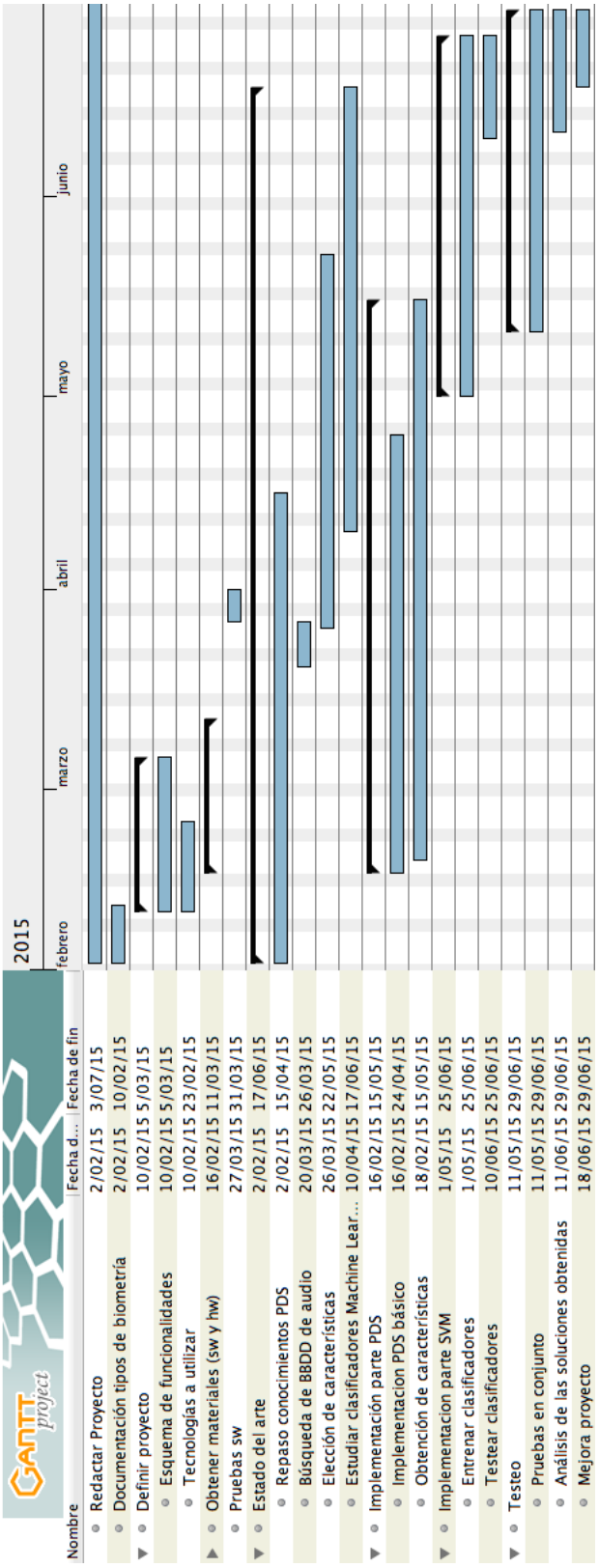


Figura 1.7: Diagrama de Gantt de tareas realizadas.

### 1.5.1. Desglose de Tareas

En el desglose de tareas hemos cogido las tareas especificadas en el diagrama de Gantt y han sido una por una analizadas y detalladas según los objetivos propuestos para cada acción, las subtareas que conllevan, los resultados esperados, fechas propuestas y el encargado o responsable que tanto las fechas como la realización de la tarea dentro de la fecha establecida se cumplan.

Tabla 1.1: Desglose Tarea 1

| **Tarea** | **1** |
|---|---|
| **Descripción** | Documentación tipos de biometría |
| **Objetivos** | Familiarizarse con los diferentes tipos de biometría y conocer las diferentes ofertas del mercado |
| **Subtareas** | – |
| **Resultados esperados** | Toma de contacto y adquisición de conocimientos |
| **Fechas** | 02/02/2015 - 10/02/2015 |
| **Responsables** | Desarrollador |

Tabla 1.2: Desglose Tarea 2

| **Tarea** | **2** |
|---|---|
| **Descripción** | Definir proyecto |
| **Objetivos** | Realizar un estudio de viabilidad |
| **Subtareas** | Esquema de funcionalidad y definir tecnologías posibles para el desarrollo e ir conociéndolas |
| **Resultados esperados** | Limitar en un rango pequeño las tecnologías a utilizar e Aproximación al objetivo de desarrollar el proyecto |
| **Fechas** | 10/02/2015 - 05/03/2015 |
| **Responsables** | Desarrollador |

Tabla 1.3: Desglose Tarea 3

| **Tarea** | **3** |
|---|---|
| **Descripción** | Obtener materiales (sw y hw) |
| **Objetivos** | Instalación de todo el software necesario para el desarrollo del proyecto |
| **Subtareas** | Puesta a punto del software necesario, incluyendo librerías |
| **Resultados esperados** | Tenerlo a punto para la fecha prevista, ya que esta tarea dependerá del desarrollo de la siguiente |
| **Fechas** | 16/02/2015 - 11/03/2015 |
| **Responsables** | Desarrollador |

Tabla 1.4: Desglose Tarea 4

| **Tarea** | **4** |
|---|---|
| **Descripción** | Pruebas sw-hw |
| **Objetivos** | Comprobar que existe una comunicación real hardware-software |
| **Subtareas** | – |
| **Resultados esperados** | Comprobar que existe una comunicación real hardware-software |
| **Fechas** | 16/02/2015 - 11/03/2015 |
| **Responsables** | Desarrollador |

Tabla 1.5: Desglose Tarea 5

| **Tarea** | **5** |
|---|---|
| **Descripción** | Estado del arte |
| **Objetivos** | Repasar conocimientos de PDS, encontrar BBDD de audio, elegir características y estudiar diferentes clasificadores. |
| **Subtareas** | – |
| **Resultados esperados** | Tener todos los conocimientos necesarios para proceder a desarrollar el proyecto de biometría de voz. |
| **Fechas** | 02/02/2015 - 17/06/2015 |
| **Responsables** | Desarrollador |

Tabla 1.6: Desglose Tarea 6

| **Tarea** | **6** |
|---|---|
| **Descripción** | Implementación parte PDS |
| **Objetivos** | Implementar el procesado básico de señal para poder tratar las señales. |
| **Subtareas** | Obtener características propias de la señal de voz |
| **Resultados esperados** | Tener las características divididas por cada locutor para poder implementar la parte SVM |
| **Fechas** | 16/02/2015 - 15/05/2015 |
| **Responsables** | Desarrollador |

Tabla 1.7: Desglose Tarea 7

| **Tarea** | **7** |
|---|---|
| **Descripción** | Implementación parte SVM |
| **Objetivos** | Entrenar y testear clasificadores |
| **Subtareas** | – |
| **Resultados esperados** | Obtener resultados para sacar conclusiones del trabajo realizado |
| **Fechas** | 01/05/2015 - 25/06/2015 |
| **Responsables** | Desarrollador |

Tabla 1.8: Desglose Tarea 8

| **Tarea** | 8 |
|---|---|
| **Descripción** | Testeo |
| **Objetivos** | Realizar las máximas pruebas posibles. |
| **Subtareas** | Mejora de aspectos analizados |
| **Resultados esperados** | Mejorar el análisis en base a los resultados obtenidos en la experimentación |
| **Fechas** | 11/05/2015 - 29/06/2015 |
| **Responsables** | Desarrollador |

## 1.6. Desarrollo

En este apartado se comenta muy brevemente qué medios han sido necesarios para el desarrollo de este sistema, con el fin de posteriormente explicar directamente cómo y para qué han sido utilizadas estas tecnologías, cual es su funcionamiento y cómo se ha progresado en el desarrollo del proyecto paso por paso.

1. Medios Software
    - Mac OS X 10.8.2
    - LaTex
    - TexMaker
    - Office 2010
        - Microsoft Project
    - Sublime Text
    - Matlab
    - Scikit
    - SVM
    - Sms-toolkits
    - Matplotlib
    - Gantt Project
    - OmniGraffle Professional

- Python 2.7
- VLC
- Adobe Audition 3.0

2. Medios hardware
   - Micrófono
   - Ordenador
3. Lenguajes de programación
   - Python
   - ScyPi
   - Matlab
   - C++

## 1.7. Conclusiones

En este capítulo se ha realizado una breve revisión de los principales avances que se han producido en los sistemas de tecnologías habladas, de biometría haciendo especial hincapié en los sistemas de biometría de voz. En primer lugar y con ánimo de presentar el escenario sobre el que se ha desarrollado este estudio se ha presentado la problemática que tiene la verificación de locutor con la Biometría de Voz y los objetivos propuesto para cumplir con el objetivo final que es crear un DNI de la voz.

A continuación en la seccion 1.3 se ha justificado el desarrollo de este proyecto, tanto social, como tecnológico y económico.Más adelante (sección 1.4) se presentan las fases por las que se ha pasado en el desarrollo de este estudio. Para concretar las fases, en el apartado 1.5 se especifican las tareas realizadas con su correspondiente desglose en el cual se concretan los objetivos propuesto.

Por último en el apartado 1.6 se comentan brevemente las tecnologías utilizadas para su desarrollo.

# Capítulo 2

# Estado del arte

A continuación se repasan brevemente los avances producidos en este campo, desde sus inicios, empezando por los sistemas que utilizan la tecnología hablada hasta llegar a los sistemas de verificación de locutor mediante la Biometría de Voz, continua con el estudio de las tecnologías necesarias para realizar este tipo de sistemas, pasando por cual es el futuro para este tipo de tecnologías y definiendo los rasgos típicos biométricos que caracterizan una señal de voz.

Por último se tratan las aplicaciones biométricas más usadas actualmente en el mercado.

## 2.1. Antecedentes históricos tecnologías habladas

El objetivo principal de la Biometría de Voz es el de facilitar a los usuarios el acceso a diferentes sistemas o a realizar diferentes tipos de tareas, tal y como se explica más adelante, además de aportar seguridad y tranquilidad debido a que la voz es un rasgo único para cada individuo.

La voz es la forma más natural de comunicación entre humanos. En este sentido, más allá de la naturalidad, el uso del procesamiento de la voz como un modo de interacción con los sistemas ofrece innumerables ventajas respecto a las interfaces clásicas de comunicación como puede ser, por ejemplo, una pantalla táctil [9].

Ha sido hace no mucho tiempo cuando se ha intentado realizar una comunicación de la misma manera con la tecnología. La Biometría de Voz es una tecnología que se encuentra en pleno desarrollo y que mucha gente todavía no conoce.

El primer paso con las tecnologías habladas fue crear una máquina parlante, creada por el fisiólogo Ch. G. Kratzenstein, en Copenhague a mediados del siglo XVII con tubos obtenidos de órganos musicales [10], aunque esta máquina aún estaba un poco lejos de la realidad. Fue en el año 1835 cuando Joseph Faber se propuso crear una máquina, Euphonia, que es la que se ve en la figura 2.1, y que consiguió que fuera fiel al habla humana.

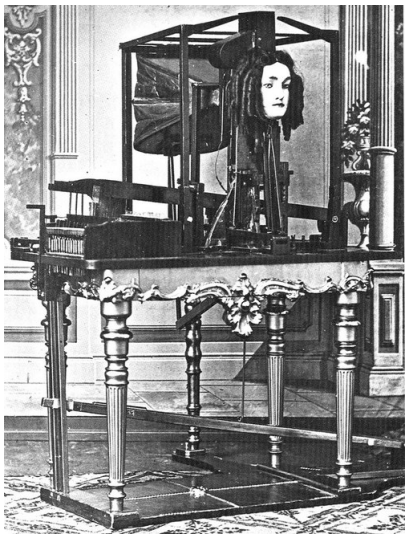

Figura 2.1: Euphonia, creada por Joseph Faber en 1835

Fue entonces cuando empezó a estudiarse la voz y a desarrollar bases matemáticas, como el análisis de Fourier. También se realizaron una serie de inventos como el fonógrafo y el teléfono que permitieron llegar más allá en el tratamiento de señales de audio [17].

Posteriormente se creó el conocido VODER, por Homer Dudley en 1939 [11], es un teclado que generaba sonidos tanto vocálicos como consonánticos. Esta fue la base de los sintetizadores de voz. En 1940 se creo el espectrógrafo de voz. Esta tecnología permite conocer la evolución de la energía de una señal de voz en diferentes bandas de frecuencia. Fue un gran descubrimiento ya que se empezó a investigar acerca del reconocimiento del habla.

El primer sistema de reconocimiento de locutor surgió en los laboratorios Bell en el año 1952. Este sistema era capaz de reconocer los números en inglés[12].

A mediados de los 60 se empieza a invertir en investigaciones para desarrollar algoritmos y realizar estudios en sistemas de reconocimiento acústico. Y aunque los resultados obtenidos no fueron los deseados se sacaron conclusiones que hoy en día han servido para

que esta tecnología avance más rápidamente [18]. El problema principal radicaba en que no se tenían computadoras digitales lo suficientemente potentes como para tratar con tanta información al momento.

En cuanto a la biometría, ha sido utilizada desde el siglo XIV en China, y a partir de finales del siglo XIX en las culturas occidentales. El precursor de la biometría es Joao de Barros, él estampaba las huellas de la palma de las manos de los niños en un papel con tinta, con el objetivo de distinguir a niños de jóvenes.

Posteriormente, Alphonse Bertillon, en 1883 desarrolló el sistema antropométrico, más conocido como Bertillonage, comentado en el apartado 1.1. Se trata de medir de forma muy precisa la anchura de la cabeza, del cuerpo y sus longitudes, y registrar si el individuo tenía algún tatuaje o cicatriz. Cuando vieron las carencias de este sistema se pasó a registrar a la gente con su huella dactilar. El objetivo era tener a los criminales identificados.

A día de hoy hay muchos tipos de biometrías, partiendo de la huella dactilar y pasando por el reconocimiento de iris, la voz, hasta la forma de caminar. Pero este estudio se centra las tecnologías del habla, en concreto en la Bimetría de Voz. Las tecnologías del habla comentadas anteriormente se pueden dividir en diferentes apartados hasta llegar al punto de verificación biométrica de voz. En la siguiente figura se ven los diferentes campos de aplicación de estas tecnologías.

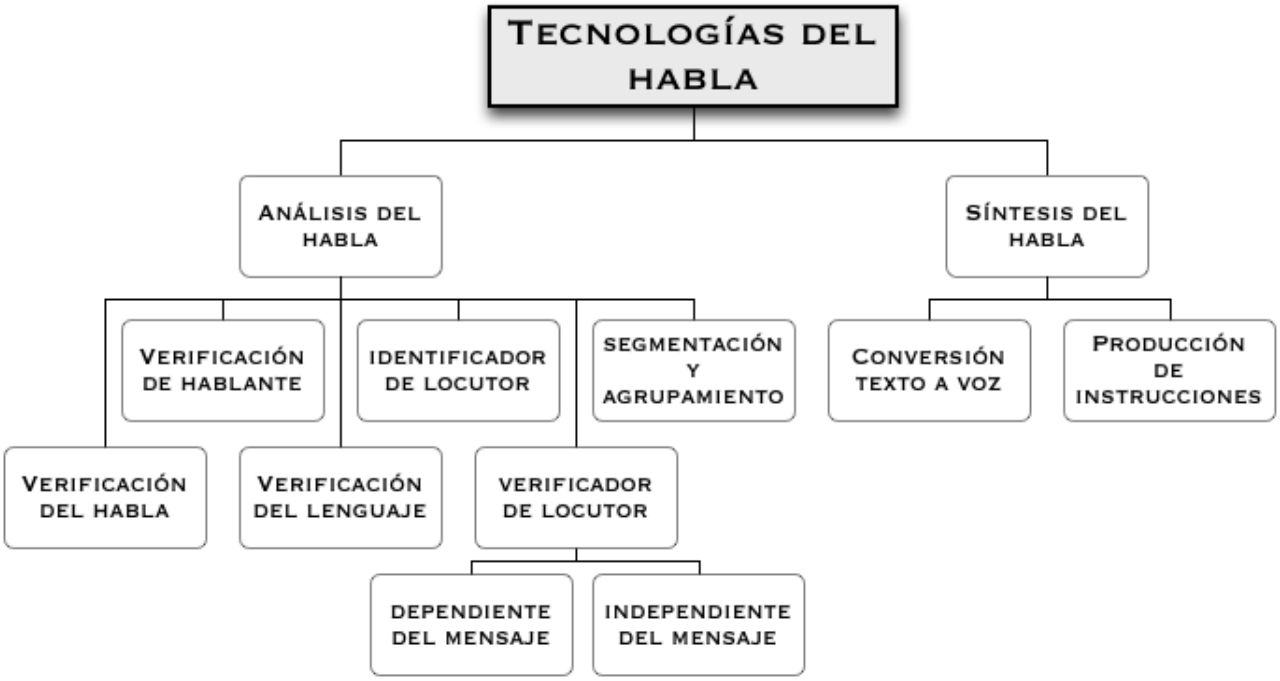


Figura 2.2: Tecnologías del habla relacionadas con la Biometría

Estas tecnologías se basan en obtener información de la señal de voz capturada. La síntesis consiste en crear una comunicación con la máquina. Pero este estudio se centra en la rama de análisis del habla, concretamente en la verificación de locutor independientemente del mensaje. Por lo que es irrelevante el idioma y sus las palabras.

En los sistema de Biometría de Voz, las entradas de voz del usuario se recogen por un micrófono y se entregan al sistema reconocedor de voz para que proporcione uno o varios resultados de reconocimiento. Estos resultados son a continuación transferidos a un sistema de procesamiento, que se encarga de extraer los resultados, en base a estos resultados y a la información derivada de los turnos anteriores (plantillas creadas), decide que estrategia tomar. Posteriormente formalizará el mensaje de respuesta del sistema a la entrada del usuario.

Es importante conocer las diferencias entre identificación y verificación de locutor, que aunque pueden parecer similares, las características a extraer y el procedimiento a seguir serán diferentes.

### 2.1.1. Identificación de locutor

En los sistemas de identificación de locutor, el propio sistema no sabe de quien se trata. Para ello parte de una base de datos de muestras de diferentes candidatos conocidos, de forma que compara estas muestras con las del hablante desconocido.

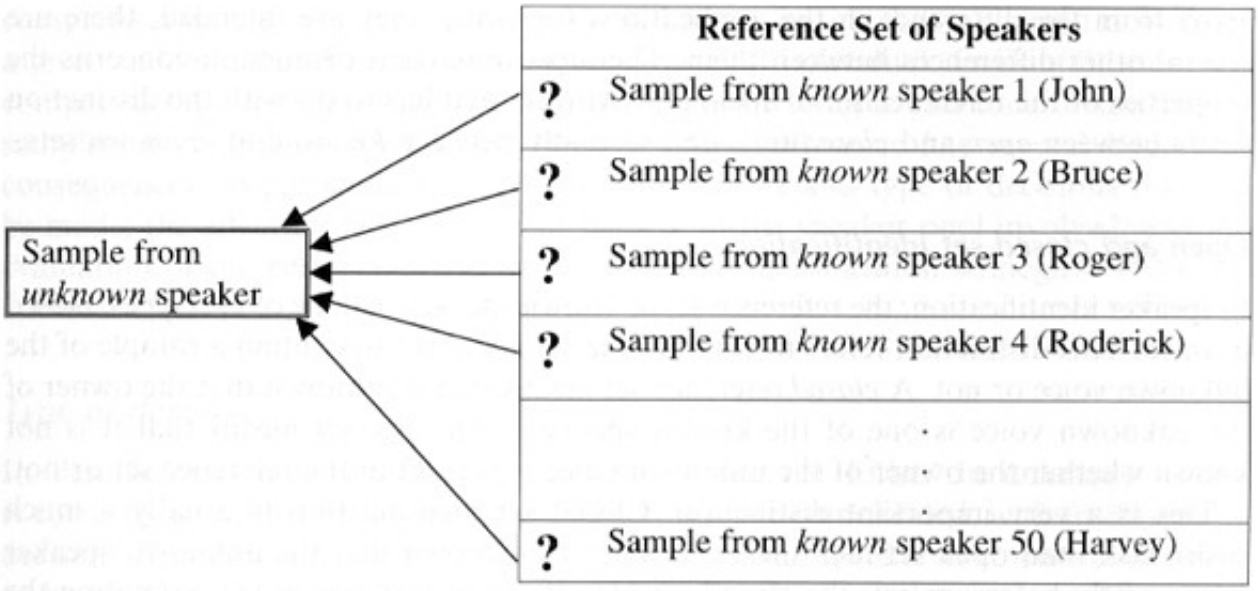


Figura 2.3: Esquema Identificación de Locutor. Rose, P. (2002) Forensic speaker identification.

El objetivo es determinar si alguna de las muestras de hablantes conocidos proviene del locutor desconocido.

En todas las biometrías es necesario tener plantillas de los locutores para poder identificar al locutor que más se acerque a las muestras analizadas.

### 2.1.2. Verificación de locutor

Tal y como se explica en el caso anterior, se compara la muestra del habla de un locutor con las muestras de un conjunto de locutores entre los que se encuentra. Cada usuario tiene un modelo de hablante además de diferentes plantillas con muestras negativas (considerándose todas las que no son del mismo usuario que se está verificando). La verificación de locutor decide si la muestra de voz capturada se corresponde con un locutor en concreto y se determina si el locutor es quien dice ser.

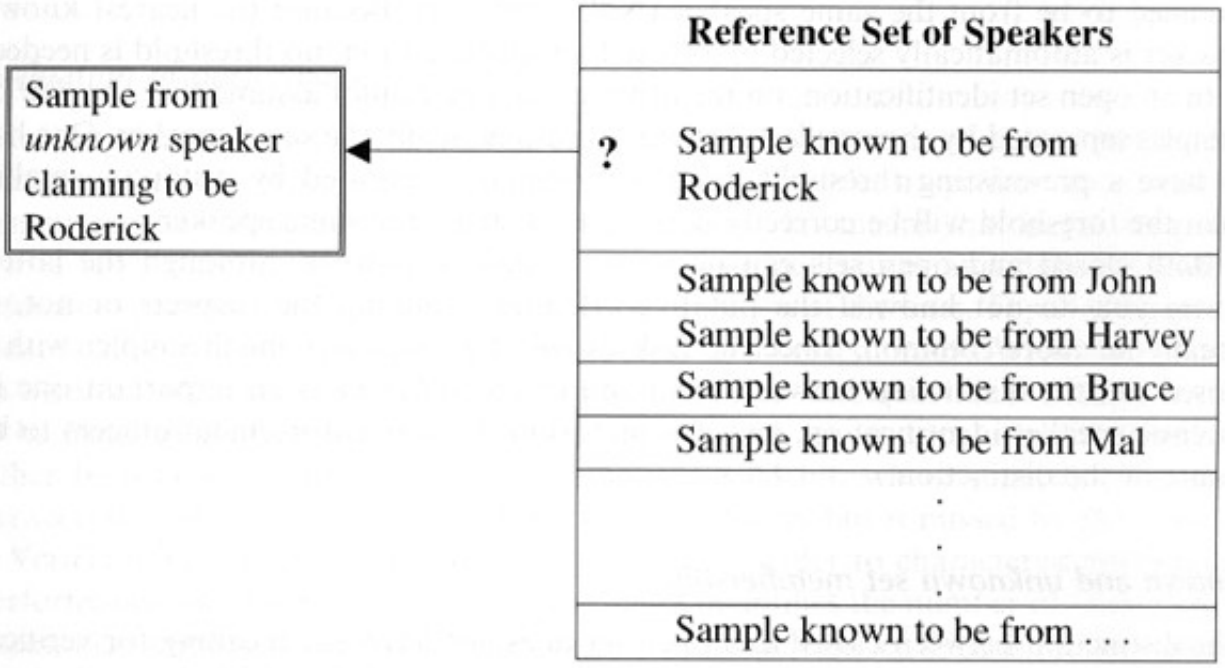


Figura 2.4: Esquema de Verificación de Locutor. Rose, P. (2002) Forensic speaker identification.

Este tipo de sistemas son muy utilizados por ejemplo en aplicaciones comerciales, transacciones bancarias y gestiones telefónicas entre otras Todas las aplicaciones persiguen un objetivo común que es aumentar la seguridad. Aunque también es muy utilizado en la acústica forense, ya que se puede utilizar en juicios comprobando si la voz presentada como prueba es de un acusado o si corresponde a otra persona [19].

Este tipo de sistema es el que hemos estudiado y desarrollado en este proyecto. En este caso también se parte de diferentes plantillas o locuciones facilitadas por voluntarios[1] las cuales serán de utilidad para el desarrollo del estudio.

En los sistemas de verificación, la comparativa se lleva a cabo por el sistema de "uno a varios". En estos casos, el sistema solo necesita conocer las características, y no la identidad. Se decidirá si el usuario está o no identificado. El tiempo de respuesta dependerá del número de entradas que tenga el sistema, esto es, de cuantos usuarios tenga.

El modo de funcionamiento de un sistema de verificación de locutor es el que vemos en la figura 2.5.

[1]Sitio web de descargas de locuciones subidas por voluntarios: voxforge.org/es/downloads

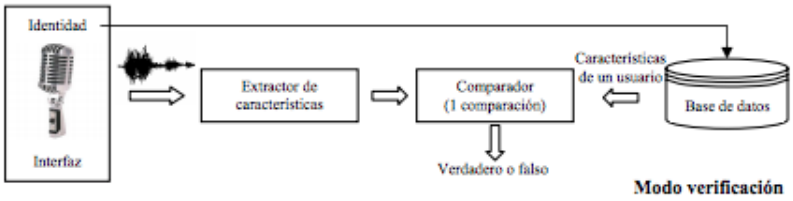


Figura 2.5: Modo de funcionamiento de un sistema de reconocimiento de locutor.

Además se pueden dividir en dos grupos:

- Dependiente del mensaje.
- Independiente del mensaje.

En los sistemas de verificación dependientes del mensaje, el usuario posee una contraseña o un código con el cual accede a una aplicación o a un lugar. Para ello se entrena al sistema con esa palabra o frase y en caso de querer cambiar de contraseña el sistema debe entrenarse de nuevo.

Por otra parte, en los sistemas independientes del texto, no es relevante entrenar al sistema con palabras o con frases concretas. El problema es que requiere mayor número de muestras y unos umbrales más estrictos.

## 2.2. Antecedentes tecnológicos

Las tecnologías utilizadas, han sido analizadas y elegidas principalmente por su condición de software libre ya que dejan amplias posibilidades para la realización de proyectos y tanto su diseño como su distribución es libre. Es decir, podemos utilizar cada una de las tecnologías libremente para el desarrollo de cualquier tipo de proyecto sin necesidad de adquirir ningún tipo de licencia.

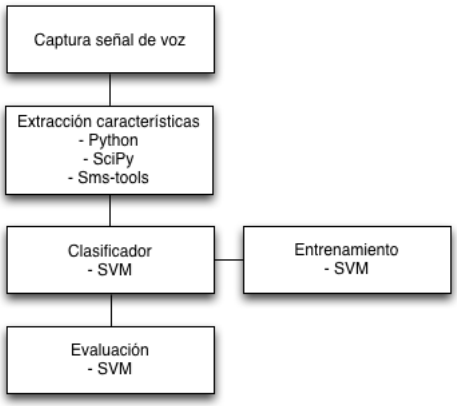


Figura 2.6: Tecnologías utilizadas para el desarrollo del proyecto.

Además de explicar cada uno de los apartados, se trata de entender porque son necesarios y cual es su función. Aunque se explica más adelante con más detalle sus funcionalidades y como se ha desarrollado el sistema Biométrico con cada una de estas tecnologías.

### 2.2.1. Python

Python fue creado por Guido Van Rossum en 1989 con el objetivo de enseñar a gente que desconocía los conceptos más complejos de la programación de ordenadores.

Es el lenguaje utilizado en este proyecto. Ha sido elegido por su condición de ser un lenguaje libre y por no ser un lenguaje complejo, tiene la filosofía de ser una sintaxis que favorezca un código legible y sencillo para el usuario. Es un lenguaje de alto nivel en claridad y simplicidad de expresión.

Python ha resultado ser un buen candidato para el desarrollo de este proyecto no solo por la simplicidad que comentamos si no también por las extensas librerías que aportan, su portabilidad y la habilidad para integrarse con otros lenguajes, como puede ser C++, que también será utilizado en el desarrollo.

Es un lenguaje de programación orientada a objetos multiplataforma implementado en C. Además aporta portabilidad con escalabilidad, velocidad de desarrollo, lo que hacen que sea un lenguaje ideal tanto para pequeñas aplicaciones como para programas muy sofisticados. Es compatible con casi todas las plataformas de hardware, y soporta todos los sistemas operativos importantes (Unix, Windows y Mac OS).

### 2.2.2. SciPy

Es una librería que ofrece herramientas matemáticas y diferentes algoritmos para utilizar con Python. También se trata de un software libre, distribuido bajo la licencia BSD desarrollado por una comunidad tutelada por Enthought. Scipy tiene como objetivo crear un entorno de trabajo similar a Matlab. Las librerías SciPy se componen de diferentes módulos pero el que resulta interesante een este estudio es el conocido como Numpy, que es el que permite trabajar con arrays multidimensionales.

### 2.2.3. Numpy

Son unas librerías de Python, una extensión que permite trabajar con vectores y con matrices partiendo de una biblioteca de funciones matemáticas de alto nivel. Fue creado por Travis Oliphant partiendo de la base que creo Jim Hugunin llamada Numeric.

Se puede definir como una calculadora muy eficiente que permite tener arrays multidimensionales y realizar operaciones muy rapidas sobre ellos, reformatearlos y estadística básica, entre otras.

Una de las ventajas de añadir esta extension a Python es que soporta más tipos numéricos que Python (como bool, int, int8-16-32-64, float,?) y convertirlos de un tipo a otro de forma más rápida.

Los principales tipos de datos que utiliza Numpy son el array y las matrices, normalmente de tipo numérico e indexados por enteros. Un array multidimensional significa que tiene muchas dimensiones o muchos ejes.

### 2.2.4. Matplotlib

Permite visualizar los datos de forma muy rápida y obtener gráficas de gran calidad. Matplotlib se divide en diferentes módulos y en concreto el que se ha utilizado en este proyecto ha sido pyplot, el cual permite dibujar graficas de forma bastante sencilla, como

si se estuviera trabajando con Matlab.

Matplotlib se utiliza en paralelo con Numpy o también de forma independiente. Además da la opción de guardar las gráficas en ficheros de tipo png, svg y pdf entre otros.

### 2.2.5. Sms-Tools

Sms-Tools son librerías creadas por el profesor Xavier Serra, de la Universidad Pompeu Fabra con las que se ha realizado el procesado básico de señal y se ha aprendido a tratar las señales acústicas de forma que se tenga buenas resoluciones en los dominios deseados para cada momento.

Son librerías pensadas para el procesado de señal en aplicaciones musicales, pero ha resultado de utilidad para recordar conceptos olvidados sobre el Procesado Digital de Señal.

### 2.2.6. Scikit-Learn

Scikit learn es un modulo desarrollado en Python con el cual podemos crear algoritmos de aprendizaje automático. En nuestro caso lo utilizaremos para desarrollar los clasificadores, para realizar los entrenamientos y para hacer las pruebas.

Clasificar significa asociar a una clase una serie de objetos, la asociación se realiza en base a las características de los objetos analizados. Para ello debemos utilizar vectores de características normalizadas.

### 2.2.7. Matlab

Matlab viene de la abreviatura Matrix Laboratoy, surgió en 1970 y es una herramienta de software matemático que ha ido creciendo con su propio lenguaje de programación (el lenguaje M) ofreciendo un entorno de desarrollo integrado. Su objetivo principal es servir de apoyo a clases o cursos de teoría de matrices, álgebra lineal y análisis numérico.

La eficiencia óptima la encontramos si trabajamos con matrices y vectores, debido a que es un programa de cálculo numérico orientado a este tipo de variables matemáticas. Aunque por otro lado la programación de Matlab está hecha en Java y Java en C++, por lo que el tiempo de procesamiento sería mayor debido a la necesidad de cambiar de Java

a C++ para que la máquina pueda interpretar el código.

La mayor desventaja que tiene Matlab es que no es un software gratuito y su licencia ronda los 1500 euros anuales. Este software se utilizó para hacer una demo inicial, por lo que realmente no ha sido necesario para el desarrollo de este proyecto.

Matlab se ha utilizado para realizar un análisis de viabilidad y demostrar de manera rápida que la creación de este sistema biométrico era posible.

### 2.2.8. Support Vector Machine

Support Vector Machine (SVM) es un método de aprendizaje supervisado con algoritmos de aprendizaje capaces de analizar y reconocer patrones. Son utilizados ampliamente en problemas tanto de regresión como clasificación. Dado un conjunto de datos de entrenamiento, cada uno perteneciente a dos clases diferenciadas, SVM es capaz de establecer una frontera de decisión que ante nuevas muestras clasifique de manera correcta a que clase pertenece. SVM se puede definir por tanto como un clasificador lineal binario no-probabilístico.

Tal y como se ilustra en la figura 2.7, SVM sigue el esquema general de cualquier método de clasificación, donde se extraen y normalizan las características a utilizar por parte del sistema. Después dichas características se utilizan para entrenar diferentes clasificadores que serán capaces de predecir de manera correcta la pertenencia a una u otra clase de la nueva muestra de entrada.

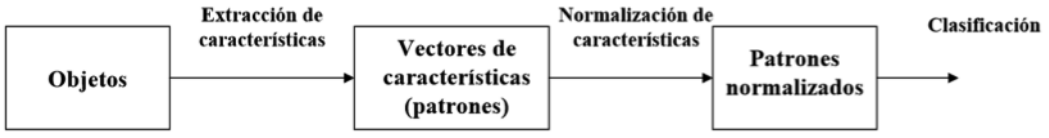


Figura 2.7: Sistema general clasificación de patrones.

## 2.3. Aplicaciones biométricas

Las aplicaciones biométricas y el reconocimiento de patrones puede venir de diferentes puntos. Vemos un esquema claro en la imagen 2.8 de las posibilidades que ofrece la

biometría en ámbitos de seguridad:

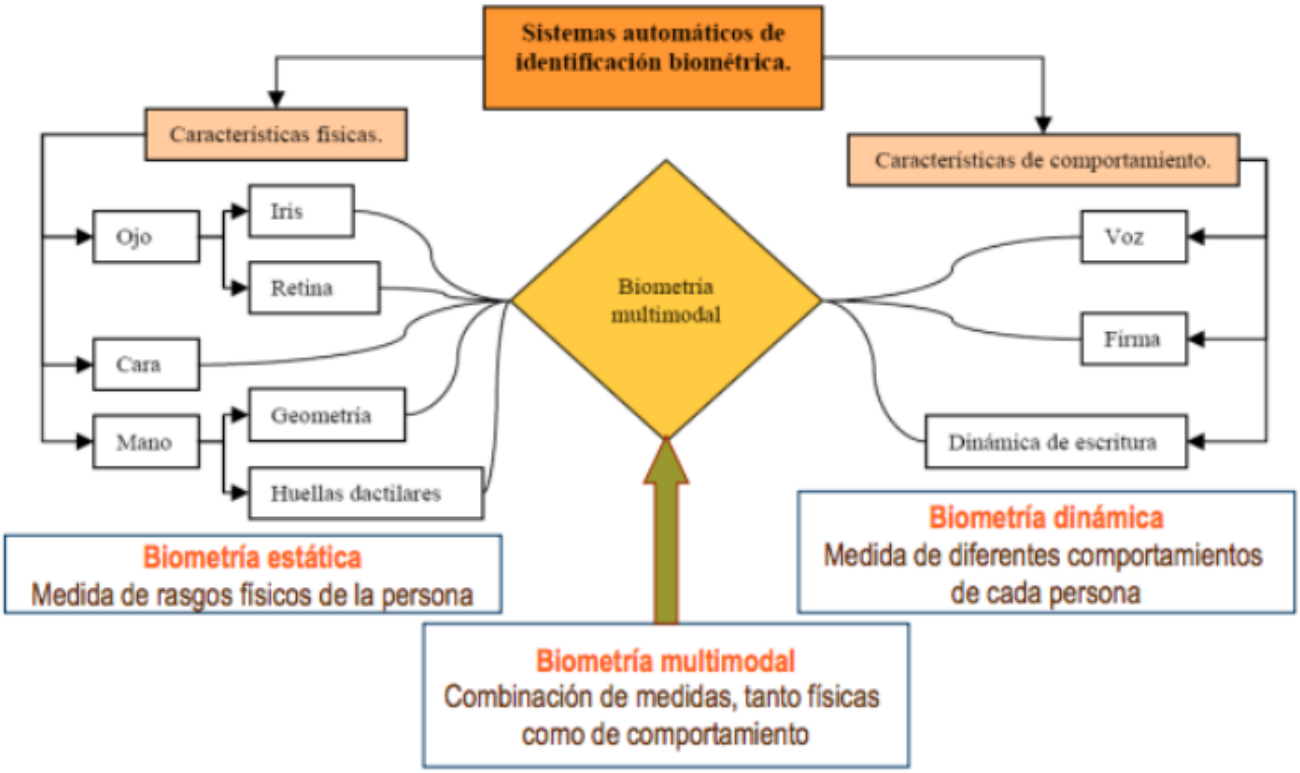


Figura 2.8: Tipos de biometrías. Criptografía, Biometría y Esteganografía, Antonio José Camacho Institución Universitaria.

Las técnicas biométricas de identificación son las que se utilizan para la identificación y/o autenticación de las personas mediante el uso de una o varias características fisiológicas de los individuos, ya sea para el acceso a algún lugar restringido o para ser identificados y de esta forma permitir o no el acceso a un sistema [**?**].

En cuanto a Biometría de Voz la tecnología ha avanzado mucho en los últimos años.

## 2.4. Conclusiones

En este capítulo se realiza una aproximación a los principales avances desarrollados en los sistemas de Biometría de Voz, dando más énfasis a los sistemas de verificación de locutor. En primer lugar, se han comentado los inicios históricos de las tecnologías habladas y de la biometría.

Posteriormente se han citado los antecedentes tecnológicos para el desarrollo de este estudio, las tecnologías utilizadas y se ha explicado muy brevemente la función que cumple cada una de ellas.

Por último se han citado diferentes aplicaciones biométricas y el proceso general que siguen.

# Capítulo 3

# Reconocimiento de voz

## 3.1. Diseño de bajo nivel

### 3.1.1. Producción de la voz

Es necesario comprender porque la voz puede ejercer como característica de un ser humano. Para ello, se debe entender que la voz es un rasgo físico aunque también lo es de comportamiento y por lo tanto, biométrico.[21]

Para que un sonido se produzca, se necesita un medio que propague las vibraciones, un cuerpo vibrante y una caja de resonancia que las amplifique. El sonido se produce por medio del aparato fonador [22]. Esta función se conoce como fonación. El aparato fonador esta formado por:

- Cavidades:
  - Infraglóticas: Diafragma, pulmones, bronquios y tráquea.
  - Glótica: Laringe (y cuerdas vocales).
  - Supraglóticas: Faringe, cavidad nasal bucal.
- Elementos articuladores:
  - Cuerdas vocales, paladar, lengua, dientes, labios y mandíbula.

En la siguiente figura se ven las partes que componen al aparato fonador y por donde entra el aire que hace que se produzcan las vibraciones.

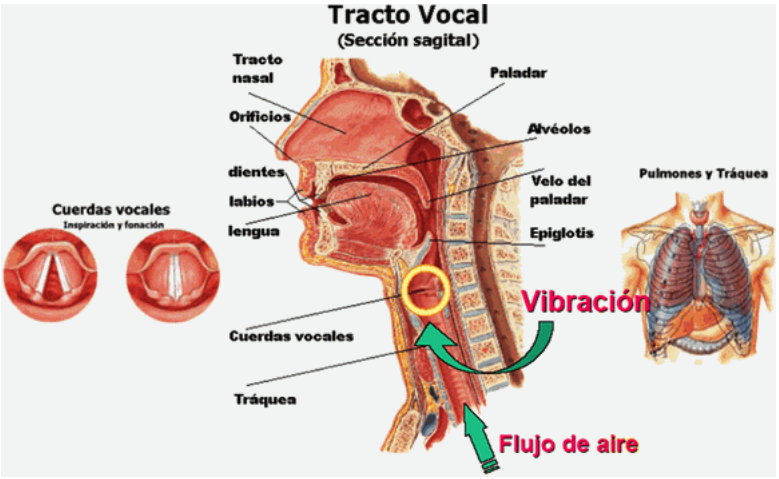


Figura 3.1: Partes que forman el aparato fonador. Vox-Technologies.

El sonido se produce en el momento en el que el aire contenido en los pulmones se expira y este aire atraviesa la tráquea haciendo vibrar las cuerdas vocales. La laringe es la encargada de generar la voz con su tono fundamental y sus armónicos.

La voz se ve modificada en la caja de resonancia ya que esta la amplifica y crea el timbre de voz. Por último los elementos articuladores moldean el aire formando los fonemas, las sílabas y las palabras [39].

Partimos de la base de que cada persona habla en un pitch, tono y volumen diferente, lo suficiente como para que el oído sea capaz de distinguir unas voces de otras.

Varios factores contribuyen a esta singularidad: el tamaño y la forma de la boca, la garganta, la nariz y los dientes, que se llaman los articuladores, y el tamaño, forma, y la tensión de las cuerdas vocales. La probabilidad de que todos estos factores sean exactamente los mismos en las personas es muy baja, casi nula.

Además, la forma de vocalizar también sirve de ayuda a la hora de distinguir a una persona: cómo se utilizan los músculos de los labios, la lengua y la mandíbula. Como se ha dicho anteriormente, el habla se produce al pasar el aire de los pulmones a través de la garganta y las cuerdas vocales, y acto seguido pasa a través de los articuladores. Diferentes posiciones de los articuladores crean diferentes sonidos. Esto produce un patrón vocal que se utiliza en el análisis.

Se debe tener en cuenta que aunque las características del sistema fonador son únicas

para cada individuo, debido a ciertos agentes externos tanto físicos como psicológicos la voz de una misma persona puede variar. Esto se conoce como variabilidad del intralocutor [41].
Y supone un problema en la Biometría de Voz, ya que por ejemplo, en la verificación por huella dactilar este tipo de cosas no ocurren. El motivo de estas variaciones son el estado de salud, el estado de ánimo y la edad entre otras. Además también pueden afectar a la verificación por voz factores externos al locutor, como el ruido externo que pueda existir en el momento de capturar la señal o que el micrófono no sea de la calidad deseada entre otros.

### 3.1.2. Rasgos biométricos

Para poder aceptar una característica como un rasgo biométrico debe cumplir las siguientes condiciones [23]:

- Universalidad: Cualquier miembro debe poseer este rasgo.
- Unicidad: Un individuo debe ser distinguido del resto por medio de este rasgo.
- Permanencia: El rasgo biométrico no debe sufrir cambios notables a lo largo del tiempo.
- Caracterización: Para poder ser utilizado como rasgo biométrico, debe ser medible de forma cuantitativa.

Estas cuatro características son básicas para que cualquier sistema biométrico funcione. Además, existen otras tres características que si se cumplen, dan calidad al sistema biométrico [28] :

- Rendimiento: Intención de obtener un sistema en tiempo real, con gran velocidad de respuesta para que resulte más cómodo de cara al usuario.
- Aceptabilidad: El sistema debe ser aceptado por los usuarios, ya que se requiere su colaboración.
- Fraude: La fiabilidad de un sistema biométrico se rige por la posibilidad existente de falsificar el rasgo a analizar.

Ningún rasgo biométricos es perfecto en todos estos aspectos. La voz, por ejemplo, tiene una alta aceptabilidad pero también dependiendo de la salud del usuario, entre otros factores, esta puede variar.

Un ejemplo en el cual puede existir una variación debido al estado de salud del locutor es una inflamación de garganta. En este caso varían las características físicas del sistema debido a la inflamación y por tanto el resultado sufre variaciones respecto al que consideramos como original.

## 3.2. Diseño de nivel medio

En esta sección se comentará en qué consiste un sistema de verificación de locutor a rasgos generales, cómo se adquieren las características y con qué clase de parámetros podemos tratar.

### 3.2.1. Proceso General de un Sistema de Verificación de Locutor

Un sistema de verificación de locutor debe estar compuesto por las siguientes fases:

- Adquisición de la señal de voz.
- Extracción de patrones o características.
- Entrenamiento (obtener muestras positivas y negativas del usuario).
- Proceso de verificación.

### 3.2.2. Adquisición de voz

Es necesario tener una base de datos de señales de voz amplia para poder realizar un número alto de pruebas y corroborar su funcionamiento.

La señal de voz hay que capturarla con la intención de almacenarla como señal digital para poder tratarla y proceder a la extracción de las características. En este estudio la toma de muestras se hace de forma local, las muestras capturadas son señales limpias ya que han sido grabadas en un estudio o descargadas de voluntarios. Debido a esto, no se han encontrado problemas como distorsiones al transmitir una señal por líneas telefónicas o problemas de ruido de la calle o de voces de fondo.

Además de las locuciones descargadas, se han realizado grabaciones con el objetivo de tener muestras de mejor calidad. Las muestras grabadas eran claras y sin ruido externo. Sin embargo, tanto para las descargadas como para las capturadas se han realizado tareas de edición de audio como eliminación de ruido y recortes al inicio y al final con el objetivo de eliminar ruidos que pudieran confundir a la hora de extraer características.

### 3.2.3. Extracción de parámetros

Una vez superado el apartado anterior, el siguiente paso es extraer los parámetros o las características. Los sistemas biométricos, basados en la extracción de características en el dominio de la frecuencia, se diferencian en la sección de la figura 3.2 marcada en rojo, ya que existe la opción de elegir diferentes características.

En la figura 3.2 se muestra un diagrama genérico de extracción de características de audio basado en el análisis espectral de frecuencia:


Figura 3.2: Ejemplo de extracción de características en frecuencia. Xavier Serra, Curso Audio Signal Processing for Music Applications.

En el capítulo 4 se detalla cada uno de los pasos seguidos en la figura anterior. Pero con la intención de explicarlo brevemente, se debe saber que esta figura indica que se parte de una señal $x[n]$, a la que se le aplica una ventana $w[n]$ y se genera el espectro utilizando la Fast Fourier Transform (FFT).

De este calculo se obtiene como resultado la magnitud y la fase espectrales de la señal.Partiendo de estas dos representaciones de la señal original se extraen las características más relevantes de la señal de audio.

Además de la posibilidad de obtener características en el dominio espectral, se pueden extraer en el dominio del tiempo. Aunque también se pueden describir de diferentes formas, como los descriptores tonales, los cuales ofrecen características relacionadas con el Pitch o con la prosodia.

Aquí se muestran algunas de las diferentes características en el dominio espectral, temporal y tonal:

- **Descriptores espectrales:** Bandas Frecuenciales Mel, Bandas Frecuenciales ERB [30], MFCC [31], Inarmonicidad [32] y Disonancia.
- **Descriptores en el dominio del tiempo:** Duración Efectiva [29], Zero Crossing Rate (ZCR) [33].
- **Descriptores tonales:** Pitch Salience Function [34], Pitch Yin FFT [35].

### 3.2.4. Características espectrales frame por frame

Casi todas las características o descriptores de un audio se ejecutan frame por frame, como los siguientes:

- Energía.
- Root Mean Square (RMS).
- Ley potencial de Stevens.
- Centroide espectral.
- Mel-frequency Cepstral Coefficients (MFCC).
- Pitch salience.
- Chroma (Harmonic pitch class profile, HPCP).

#### Energía

La energía de un frame de audio se puede sacar partiendo del espectro y desde el dominio del tiempo. Si estamos con el dominio espectral lo haremos sumando todos los cuadrados de sus magnitudes, tal y como indica la ecuación 3.1.

$$Energia = \sum_{k=0}^{N-1} |X_l[k]|^2 \tag{3.1}$$

### Root Mean Square

Root Mean Square (RMS), es una versión modificada de la energía. Es otra forma de visualizar la energía de la señal. Se obtiene elevando al cuadrado el valor de la perturbación de presión sonora en cada momento. Estos valores cuadráticos son sumados y promediados en el tiempo. La presión sonora RMS es la raíz cuadrada de esta media temporal.

$$RMS_l = \sqrt{1/N^2 \sum_{k=0}^{N-1} |X_l[k]|^2} \tag{3.2}$$

### Ley potencial de Stevens

En 1930, Richardson y Ross [36], observaron una relación exponencial entre las estimaciones sonoras y la presión sonora, y esta ley, predice que los cambios en la magnitud del estímulo producen cambios equivalentes en la sensación. Es una medida muy simple de cuan alto está un sonido.

Se calcula con la siguiente ecuación:

$$\sum_{k=0}^{N-1} |X_l[k]|^2 \tag{3.3}$$

### Comparativa características

En la figura 3.3 se aprecian las tres características anteriores para un sonido de un piano. Se tiene el análisis en tiempo, y se analiza cada característica para cada uno de los frames.

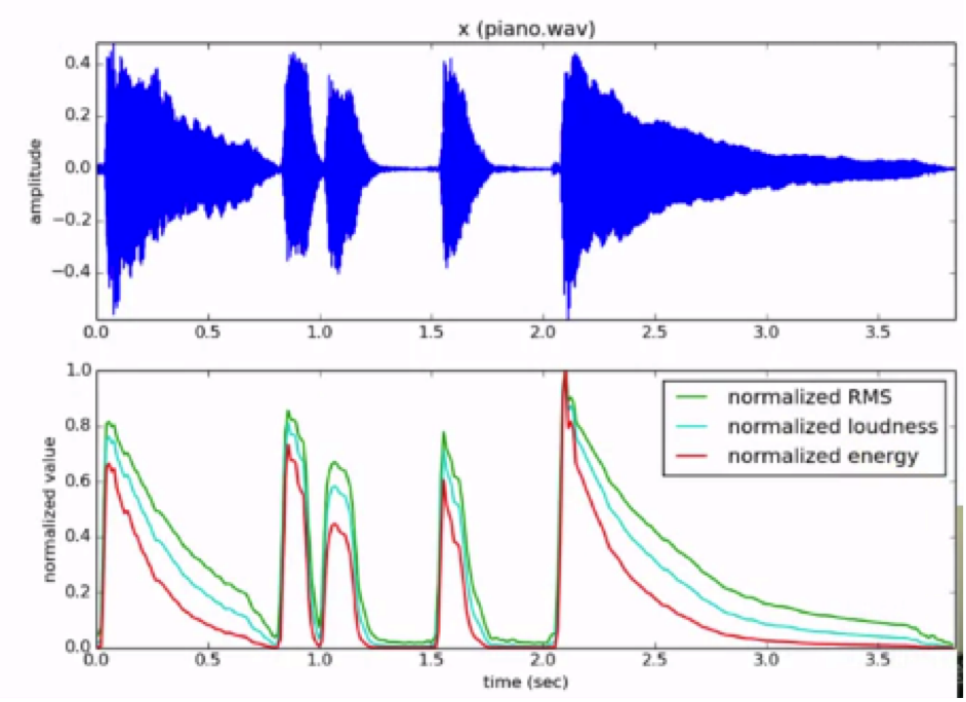


Figura 3.3: Señal acústica de una frase tocada con un piano y comparativa de características calculadas. Ejemplo realizado con Python y librerías SMS-Tools.

### Centroide Espectral

Es una característica que trata de definir la forma espectral de un sonido. Indica dónde está la parte más concentrada de espectro.

Perceptualmente esta relacionado con la claridad que puede tener un sonido y se calcula como la media ponderada de las frecuencias presentes en la señal:

$$centroide_l = \frac{\sum_{k=0}^{N/2} k|X_l[k]|}{\sum_{k=0}^{N/2} |X_l[k]|} \tag{3.4}$$

Suma el espectro completo ponderándolo por cada frecuencia y normalizándolo con la energía total. No es la suma total de la energía, si no que es la suma de las amplitudes espectrales, ya que si se suman los cuadrados de las amplitudes se obtiene la energía de la señal.

En la figura 3.4 se ve la señal de voz de un hombre con el centroide espectral calculado.

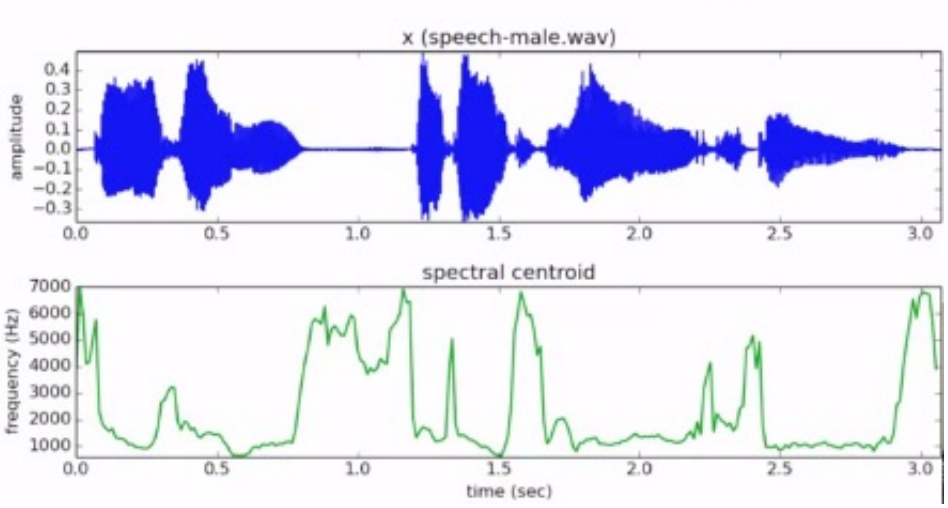


Figura 3.4: Señal de voz de un hombre con centroide espectral. Ejemplo realizado con Python y librerías SMS-Tools.

En la gráfica del centroide espectral, en la figura 3.5 se aprecia como esta señal varia de 1 kHz a 7 kHz siendo los silencios los que tienen el centroide más alto.

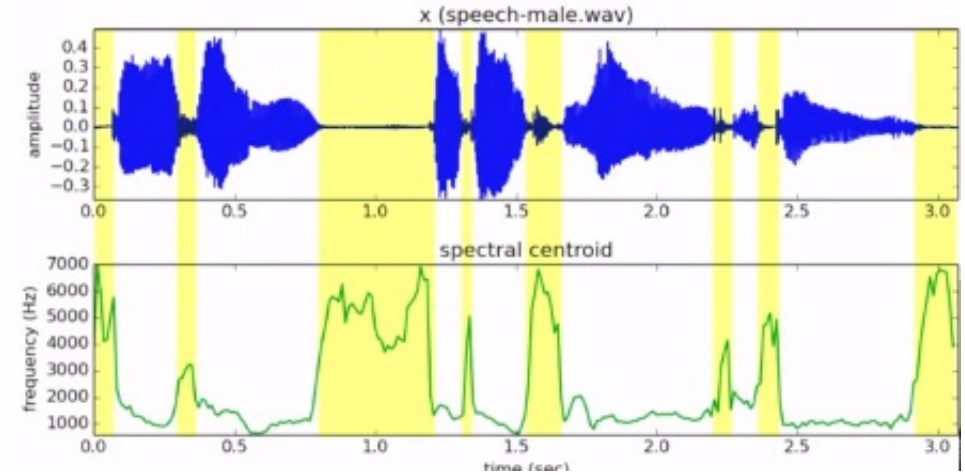


Figura 3.5: Comparativa silencios de la señal de voz con centroides

Y las partes de voz con amplitudes más altas tienen un centroide más bajo. Como se decía anteriormente, esta característica da información acerca de la claridad de una señal de audio, por lo que tiene sentido que sea así.

### Mel Frequency Cepstral Coefficients (MFCC)

MFCC es una característica que tiene que ver con la forma del espectro pero de una manera más compleja.

Es una representación de la magnitud espectral y se resuelve partiendo la transformada del coseno del logaritmo de su magnitud espectral en una escala no lineal, es la llamada escala Mel:

$$mfcc_l = DCT(log_{10}(\sum_{k=0}^{N/2} |X_l[k]|H_l[k]))$$

$$\text{donde} \begin{cases} \|X_l[k]\| \quad \textit{es la parte positiva de la magnitud espectral} \\ H_l[k] \quad \textit{es el banco de filtros de la escala de Mel} \\ DCT[m] = \sum_{n=0}^{N-1} f[n]cos(\pi/N(n+1/2)m) \end{cases} \tag{3.5}$$

La ecuación 3.5 muestra como coge el espectro completo $x_l[k]$ y lo multiplica por un banco de filtros, esto es, por una ventana. De forma que cada frecuencia que sea dependiente de la escala Mel cambie. El objetivo es hacer más perceptiva la magnitud espectral del resultado de la FFT. Después se hace el logaritmo y por ultimo la DCT.

En la figura 3.6 se encuentra el diagrama de bloques para realizar la MFCC:


Figura 3.6: Diagrama para realización de MFCC

Se parte de la magnitud espectral y se divide esta magnitud espectral con el banco de filtros o banco de porciones del espectro de acuerdo a la escala de Mel. Después se hace el logaritmo y por último la DCT, para llegar a los coeficientes MFCC [43].

La extracción de coeficientes MFCC es la técnica de parametrización más utilizada en el área de verificación de locutor. El objetivo es tener una representación robusta para tener un modelo preciso del locutor.

**Escala Mel**

La escala Mel, aparece en la figura 3.7, aproxima la resolución frecuencial al oído humano. Relaciona la frecuencia percibida (eje y) con la frecuencia real (eje x). Los humanos somos capaces de notar pequeños cambios en bajas frecuencias más fácilmente que en altas frecuencias. Utilizando esta gráfica, se consigue que las características espectrales sean más cercanas a lo que un oído humano escucha [42].

$$mel = 2595log_{10}(1 + \frac{f}{700}) \tag{3.6}$$

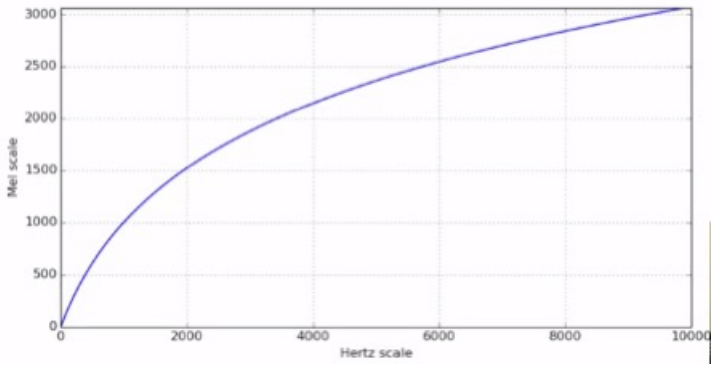


Figura 3.7: Frecuencias Mel

El eje horizontal es la escala lineal y el vertical es la nueva escala, que como se ve pone mayor énfasis en las altas frecuencias que en las bajas.

La figura 3.8 es una visualización de un análisis MFCC. Es la voz de un hombre con sus coeficientes MFC. Cada coeficiente es una representación de un nivel diferente de la forma del espectro por eso no resulta una representación muy intuitiva.

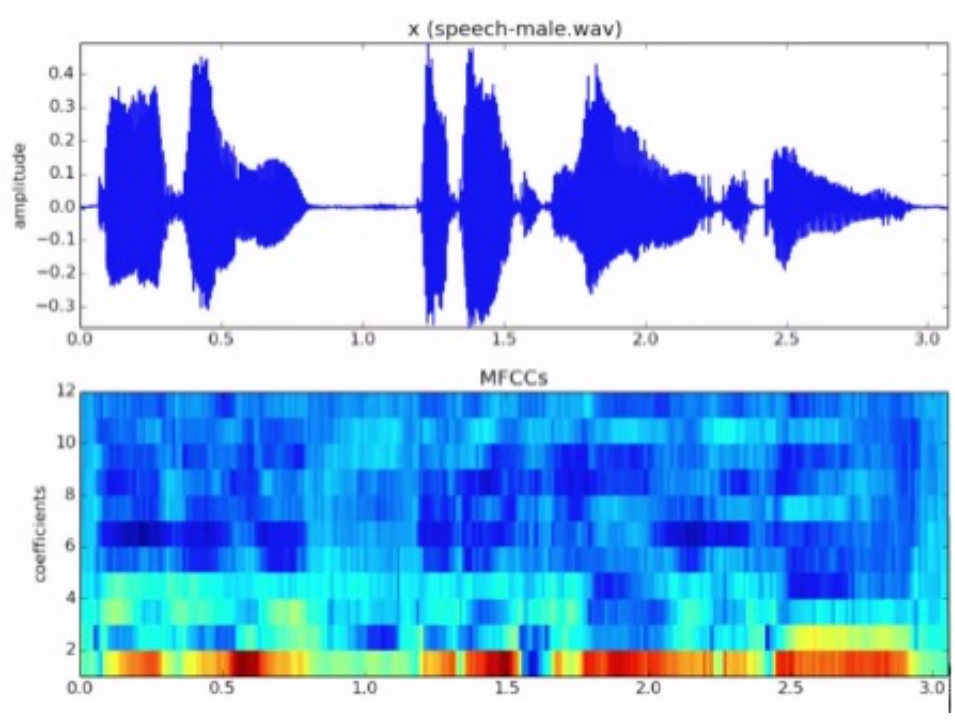


Figura 3.8: Ejemplo señal de voz con el cálculo de los coeficientes MFCC. Ejemplo realizado con Python y librerías SMS-Tools.

En la figura 3.8 se aprecian doce coeficientes (el número de coeficientes a calcular y a mostrar se puede elegir). El coeficiente número cero que no se muestra en la imagen, está relacionado con el nivel de la señal, representa la energía de la señal.

El primer coeficiente es el que describe la figura más grande del espectro, esto es, la que define la forma general del espectro y cuando vamos más arriba, da más detalles, mas cambios en el espectro. El resultado obtenido de un análisis MFCC es un vector que incluye todos los coeficientes en cada frame.

En MFCC las bandas de frecuencia están espaciadas logarítmicamente según la escala Mel, con el fin de modelar la respuesta auditiva humana de manera más apropiada que las bandas espaciadas linealmente, como en el caso de la Transformada de Fourier.

Más adelante, en el apartado 4.4 se detalla paso por paso el proceso a seguir para obtener este tipo de coeficientes.

### 3.2.5. Ejemplo de extracción de características en frecuencia

En este ejemplo se analiza un instrumento de cuerda, concretamente una frase tocada con un violonchelo. Para poder obtener características en el espectro frecuencial es importante elegir un tamaño de ventana y de salto adecuados. En este caso los valores que a utilizar serán los siguientes:

- Tipo de ventana: Blackman
- Tamaño de ventana: 1001
- Tamaño FFT: 1024
- Tamaño de salto: 250

En este proyecto se ha realizado este mismo análisis con las diferentes señales de voz con el objetivo de tener unos valores generales válidos para todas las señales que permitan obtener una buena resolución frecuencial.

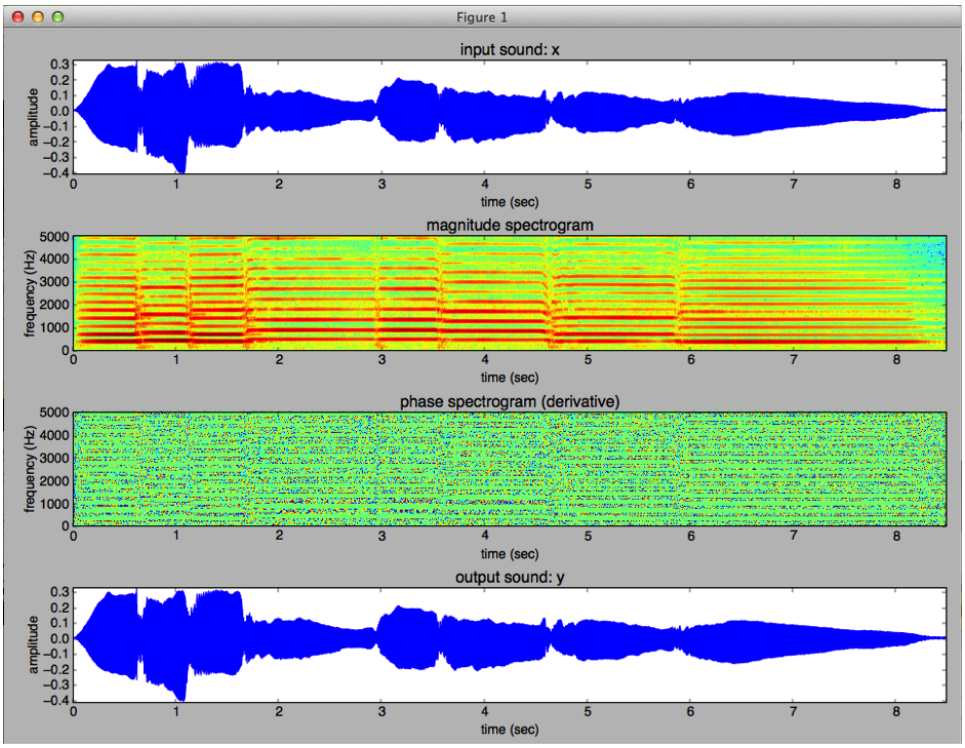


Figura 3.9: Análisis STFT de frase tocada con un violonchelo. Ejemplo realizado con Python y librerías SMS-Tools.

En la figura 3.9 se ve el análisis de la señal realizado. Por un lado se aprecia la señal original, su magnitud espectral, la fase espectral y por último la señal reconstruída. Lo que se debe hacer es observar si existe una buena resolución en frecuencia, ya que se deben extraer características en el dominio frecuencial.

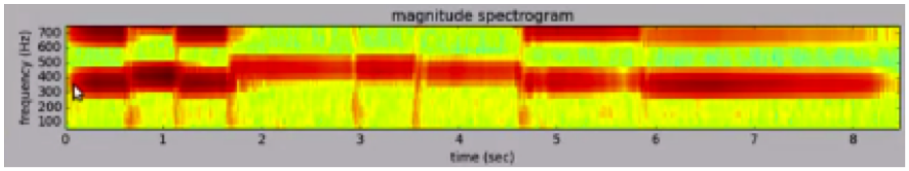


Figura 3.10: Magnitud espectral del violonchelo. Ejemplo realizado con Python y librerías SMS-Tools.

En la figura 3.10 no se puede apreciar detalle en el eje horizontal, esto es, en el eje frecuencial, no es demasiada la información aportada por estos valores en el análisis. Sin embargo, si se cambia el tamaño de la FFT a 4096, se puede hacer zero-padding con el fin de obtener más muestras por frecuencia.

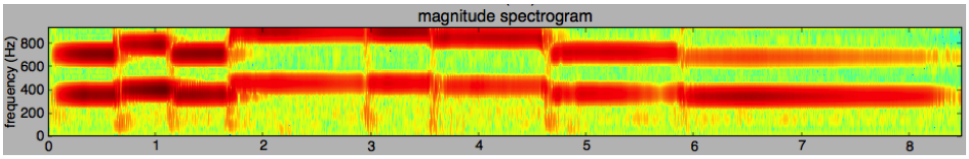


Figura 3.11: Magnitud espectral del violonchelo. Ejemplo realizado con Python y librerías SMS-Tools.

Ahora en la figura 3.11 se ve como existe más resolución en frecuencia que antes.Las líneas horizontales tienen más detalle, esto es, tenemos más información frecuencial. La frecuencia más baja está a 348 Hz aproximadamente, y la más alta a 456 Hz. Esta información sirve para decidir el tamaño de ventana a utilizar. Con las siguientes operaciones se puede calcular:

```
In [4]: 6 * 44100 / 340
Out[4]: 778.0
```

Por lo que los nuevos valores para el análisis serán los siguientes:

- Tipo de ventana: Hamming
- Tamaño de ventana: 778
- Tamaño FFT: 4096
- Tamaño de salto: 220

## 3.3. Diseño de alto nivel

### 3.3.1. Clasificación de características biométricas

Hay diferentes formas de realizar la clasificación de las características extraídas de una señal de voz, o de cualquier tipo de patrón característico y discriminante.

En los siguientes apartados se muestran las diferentes posibilidades que se han encontrado a la hora de desarrollar este estudio, centrándose y explicando más a fondo la opción que ha resultado decisiva en este proyecto, que es el uso de Support Vector Machines como método de clasificación para las características de voz extraídas.

**Descriptores del sonido** .

En la tabla 3.1 se muestra la clasificación que se ha hecho de los sonidos.

Tabla 3.1: Descriptores del sonido

| **Perceptual** | **Sensorial** | **Física** |
|---|---|---|
| intervalos sucesivos o simultáneos | pitch | frecuencia |
| tiempo (beat) | tiempo | duración |
| timbre (envolvente espectral) | timbre | espectro (centroide) |
| dinámica | volumen | intensidad |

Por un lado, la primera fila son los diferentes niveles de abstracción, partiendo de la parte física, que es la más baja con la cual se va a trabajar hasta el nivel cognitivo, que será el nivel más alto.

En el nivel físico se tratan conceptos como la frecuencia, duración del sonido, el espectro o la intensidad. Un nivel por encima está el nivel sensorial, en el cual en lugar de la frecuencia se habla del pitch, en lugar de la duración se habla del tiempo, y en lugar del espectro se habla del timbre y por ultimo en lugar de intensidad se habla del volumen.

El nivel perceptual trata conceptos más musicales, en este caso se habla de intervalos simultaneos de pitch, conocidos como nodos, en lugar de tiempo se habla del beat,y en lugar del timbre se habla de aspectos del timbre que se pueden identificar y caracterizar con aspectos musicales como la forma del espectro. Por ultimo en lugar del volumen, se habla de la dinámica de un sonido.

El sonido se debe analizar desde sus descripciones más bajas. Por lo que si se quieren describir sonidos de manera general, se pueden agrupar características del audio en diferentes categorías. En este caso se han agrupado categorías relacionadas con el timbre (centroide espectral, MFCC,..), características relacionadas con la dinámica (volumen en una señal acústica particular, el nivel medio,...) y características relacionadas con el pitch.

Partiendo de estos descriptores se dividen las colecciones de sonidos:

- Similitud entre los sonidos
- Clustering de los sonidos

- Clasificación de los sonidos

El más importante es la similitud entre los sonidos. Una vez que se haya hecho esto, se pueden agrupar sonidos teniendo en cuenta un criterio. Y finalmente, si se tienen las clases ya marcadas, se pueden clasificar los diferentes sonidos, esto es, podemos asignar clases a un sonido particular.

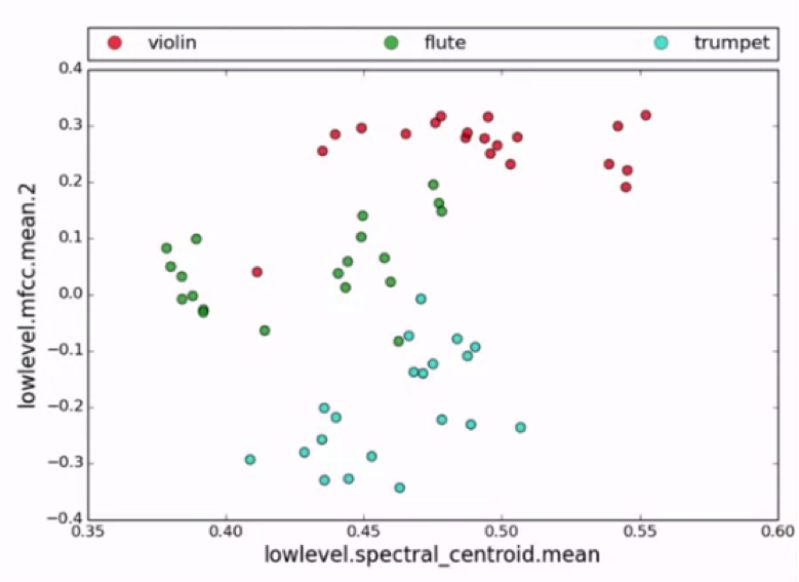


Figura 3.12: Colecciones de sonido, división violín, flauta y trompeta. Ejemplo realizado con Python y librerías SMS-Tools.

En este contexto, una colección de sonido se puede representar por un diagrama como el de la figura 3.12. Se considera un sonido como un grupo de características. Cada característica tiene un valor numérico. En el ejemplo solo se han tomado dos características. Al considerar representar un sonido simplemente con dos características, puede ser visualizado en un espacio de dos dimensiones, que es lo que se ve en la figura anterior. Cada característica es una dimensión. El eje horizontal es el centroide espectral y el vertical es el MFCC. Se han analizado las notas de 3 instrumentos, el violín, la flauta y la trompeta, se ha calculado el centroide espectral y la MFCC de la señal.

Se aprecia como el violín tiene valores más altos para los coeficientes MFCC y tiene un centroide espectral que cubre más superficie que el resto. Por otro lado, la trompeta tiene coeficientes MFCC más bajos y la flauta se encuentra entre los dos. Pero se puede comprobar que estos tres sonidos son bastante diferentes analizando estas características [44].

### Similitud entre los sonidos

**Distancia Euclídea** .

Para poder trabajar con estas clases, o con estos espacios, es importante saber medir la distancia entre los sonidos, esto es, entre los puntos. Por eso, se debe conocer la manera de, en un espacio multidimensional, comparar dos sonidos.

Para eso se utiliza la distancia Euclídea, que es la forma más sencilla de medir distancias entre dos puntos en un espacio multidimensional. En este caso, "p"sería uno de los sonidos analizado, y "q"se correspondería con otro sonido.

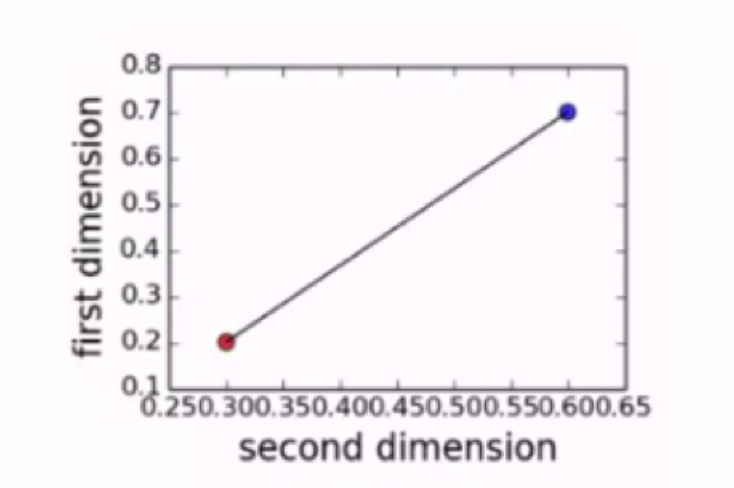


Figura 3.13: Ejemplo medición distancia Eculídea 2D.

Posteriormente, para cada dimension "i", se calcula la distancia entre los dos valores. La fórmula para calcular esta distancia es la siguiente:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \qquad donde \quad \begin{cases} p = (p_1, p_2, ..., p_n) \\ q = (q_1, q_2, ..., q_n) \\ son\ dos\ puntos\ en\ un\ espacio\ Euclídeo\ n. \end{cases} \tag{3.7}$$

En caso de tener dos dimensiones, 2D, esto es, un espacio de solamente dos características, esta tarea es mucho más sencilla.

### 3.3.2. Clustering

#### K-Means

K-Means es uno de los más simples métodos de aprendizaje no supervisado para resolver el problema de clusterización. El método que sigue dicho proceso está basado en la definición, a priori del número de K clusters en los cuales se quiere dividir la muestra de entrada.

La idea principal se basa en la definición de K centroides, uno por cada cluster. Dichos centroides deberán situarse de manera alejada los más alejados unos de otros en el instante inicial de la ejecución del método. El siguiente paso consiste en calcular las distancias de cada muestra a los centroides definidos y asociar dichas muestras al que se encuentra más próximo. Una vez se ha llevado a cabo dicho proceso, se debe calcular los baricentros de los centroides de los clusters obtenidos en el paso anterior. Cuando se dispone de dichos centroides, se recalcula la distancia de cada uno a las muestras de entrada al sistema con el fin de reasociarlas a los nuevos clusters. Este proceso se repite hasta que los centroides permanezcan estáticos en un punto debido a que se ha encontrado el mínimo coste de asociación para cada cluster.

En la figura 3.14 se observa como para una muestra de entrada, se llevan a cabo diferentes pasos del proceso de K-Means hasta obtener el mínimo coste de asociación para los dos clusters definidos.

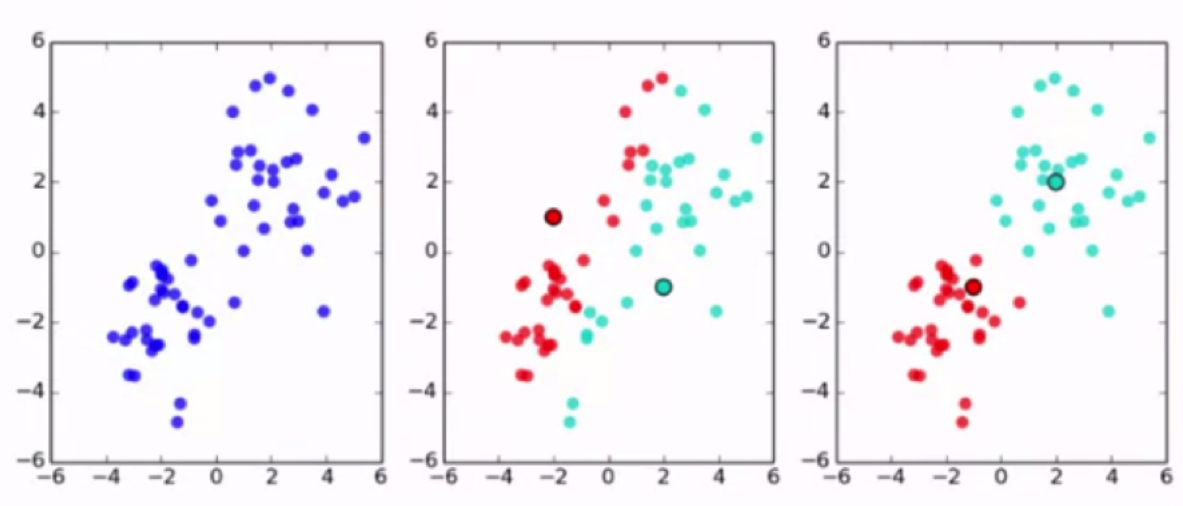


Figura 3.14: Ejemplo clasificación de sonidos con algoritmo K-means. Ejemplo realizado con Python y librerías SMS-Tools.

### 3.3.3. Clasificadores

El siguiente paso es la clasificación de sonidos. Esto significa que tenemos varias clases identificadas en diferentes categorías y lo que queremos hacer es clasificar un sonido nuevo en una de esas clases ya existentes. Un esquema típico de reconocimiento de patrones es el que vemos en la siguiente figura:

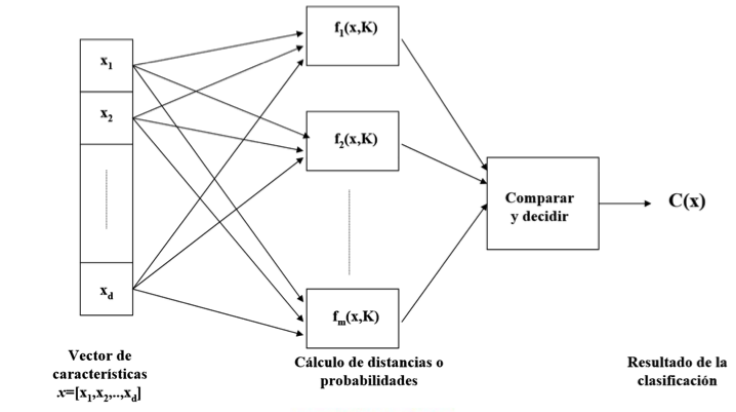


Figura 3.15: Esquema general del proceso de reconocimiento de patrones.

#### K Nearest Neighbors

K Nearest Neighbors (KNN) es un método no paramétrico utilizado en la regresión y la clasificación. Está basado en la pertenencia de una muestra de entrada a una clase con K muestras de entrenamiento más cercanas. Esto significa que cuando una nueva muestra entra al clasificador con el fin de ser asociada a una clase, se calculan las distancias a todas las muestras de entrenamiento del clasificador y se coge aquellas K muestras que se encuentren más cercanas. La clase que disponga de mayor número de K muestras más cercanas, es a la cual será asignada la muestra de entrada.

Este proceso puede ser observado en la figura 3.16 en la cual se lleva a cabo el proceso de clasificación de las muestras de color verde. Una vez calculadas las distancias a cada muestra de entrenamiento de ambas clases, se ordenan en función a dicha distancia con el fin de sacar aquellas K muestras más próximas. De esta manera obtenemos la clase a la cual pertenece las muestras de color verde.

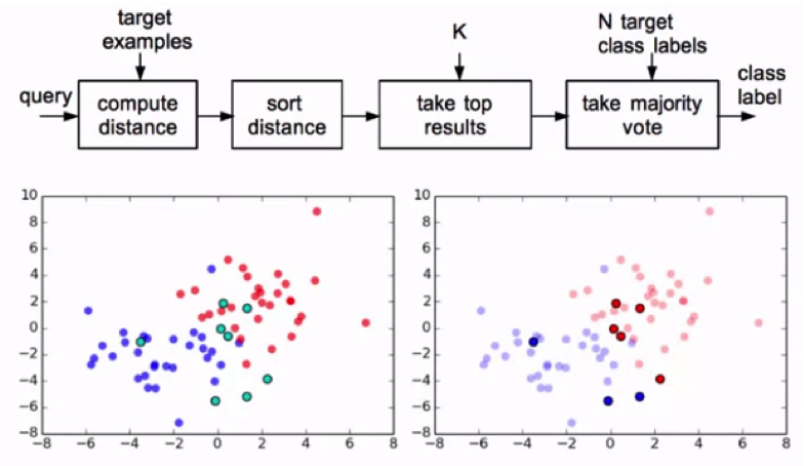


Figura 3.16: Esquema general del proceso de clasificación mediante KNN (Hastie, Tibshirani and Friedman - Elements of statistical learning).

Dependiendo del valor asignado a K, la frontera de decisión que se establece en el clasificador KNN queda mucho más a merced de los posibles espurios que puedan existir en la muestra de entrenamiento. En la figura 3.17 se observa que cuanto mayor es el valor de K, las fronteras que se establecen son más generalistas y no quedan sesgadas por la muestra de entrenamiento, tal y como se puede observar en caso de que K sea igual a 1.

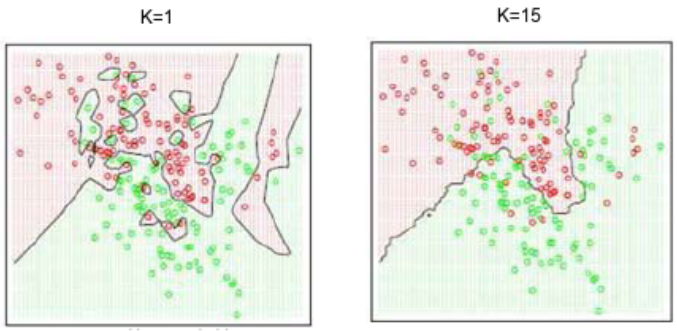


Figura 3.17: Consecuencias de diferentes valores de K (Hastie, Tibshirani and Friedman - Elements of statistical learning).

### Redes Neuronales

Las redes neuronales es un método de clasificación compatible dentro de un sistema basado en características biometrías. Este método de clasificación se basa en emular las neuronas que realizan las conexiones estructurales del cerebro humano y que permite llevar a cabo acciones tan comunes como memorizar o asociar conceptos teniendo en cuenta la experiencia adquirida por la persona.

Hay ciertos casos en los que los problemas no pueden expresarse o resolverse a través de algoritmos. Es en esos casos donde el uso de redes neuronales entra en juego. Este tipo de clasificador implementa un modelo artificial y simplificado del cerebro humano, capaces de adquirir conocimientos a través de la experiencia.

Mediante un entrenamiento de las diferentes capas que modelan un clasificador de redes neuronales, cada una de las neuronas es capaz de establecer su frontera de decisión en base a los parámetros de entrada y su correspondiente salida. Una vez modelada el conjunto de fronteras de decisión de la red de neuronas, el sistema, en base a una entrada de características es capaz de diferenciar entre el tipo de resultado que se desea como óptimo.

Este tipo de clasificador es muy utilizado en reconocimientos de palabras y se fundamenta en el reconocimiento del habla y no de los locutores. Por lo tanto la clasificación basada en el uso de redes neuronales no es una técnica de clasificación que vaya a servir de utilidad en la verificación de locutor.

**Modelos Ocultos de Markov (HMM)**

Son modelos estadísticos utilizados en el reconocimiento de locutor dependiente de texto. Este método de clasificación se basa en el modelado estadístico de un sistema, el cual asume que el sistema es un proceso de estados de Markov ocultos.

Los diferentes estados en los cuales se puede encontrar el sistema en cada momento se encuentran de manera oculta y solo los posibles resultados del sistema son visibles. Dependiendo de las diferentes transiciones entre los estados ocultos de cada clasificador en la fase de entrenamiento, dichos estados modelan una distribución de probabilidad hacia los posibles resultados.

Esto permite modelar, por ejemplo, las diferentes transiciones que pueden tener palabras con un número acotado de sílabas, donde las transiciones hacia los estados pueden ser los diferentes conjuntos de sílabas que conforman una palabra.

De esta manera, creando un diccionario de silabas y palabras, somos capaces incluso de predecir a que palabra se refiere un usuario en caso de el reconocimiento inicial haya fallado.

#### Modelos de mezclas Gaussianas

El modelado estadístico de clases mediante el uso de modelos de mezclas de distribuciones normales o gaussianas es un modelo probabilístico que asume que las distintas distribuciones de datos sobre las cuales se pretende llevar a cabo la clasificación conforman distribuciones gaussianas independientes.

Este modelo de clasificación es comunmente denominado como una clusterización de clases que lleva a cabo k-medias. Considerando que la mayoria de distribuciones de datos pueden modelarse como una distribución gaussiana con una media y una covarianza, este sencillo método es capaz de establecer las fronteras de decisión de manera sencilla. Aún así, y debido a ese factor de modelado del conjunto de datos como distribuciones normales, este algoritmo no tiene cabida en caso de que nuestros datos no conformen distribuciones de este tipo.

### 3.3.4. Support Vector Machine

El clasificador Support Vector Machines (SVM) es uno de los más utilizado en la actualidad debido a las ventajas que ofrece, como su efectividad en grandes espacios dimensionales de características, en su uso con respecto a otros. SVM está principalmente orientado a la resolución de problemas biclase, aunque las diferentes extensiones del método permiten también su uso para la resolución de clasificaciones multiclase [50]. La mayor ventaja del uso de este clasificador reside en su capacidad de maximizar la frontera de decisión entre las diferentes clases. Esta separación óptima, aún disponiendo de escasas muestras para llevar a cabo el entrenamiento del clasificador, se realiza máximizando dicha separación entre clases, haciendo uso de vectores soporte. Estos vectores soporte no son más que características que se encuentran en el extremo de la distribución de una clase y que sirven como referencia para delimitar la distancia interclase. Este concepto es fácilmente ilustrable a través de la figura 3.18.

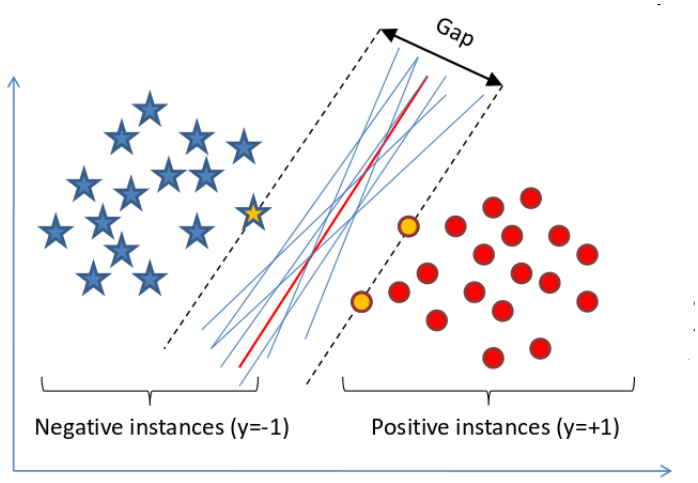


Figura 3.18: SVM: Separación biclase a traves del uso de vectores soporte

Para llevar a cabo la clasificación de manera que la frontera de decisión sea lineal, SVM utiliza el *Kernel trick* mediante el cual eleva los vectores de características a dimensiones en las cuales son linealmente separables. Aún así existen diferentes tipos de características que, aún elevándolas a un espacio dimensional muy superior al que se encuentran, no son separables mediante un kernel lineal. Por tanto existen diferentes tipos de kernels que dotan de fronteras de decisión de un orden superior. El kernel polinomial, el gaussiano o el exponencial son alguno de ellos. A continuación enumeramos las ventajas y desventajas del uso de SVM para resolver el problema planteado en nuestro trabajo:

Ventajas:

- Efectiva en grandes espacios dimensionales.
- Efectiva en casos en los que el numero de dimensiones es mayor al numero de muestras.
- Clasificador entrenable con gran velocidad de entrenamiento.
- Permite construir hiperplanos óptimos de separación en problemas de clasificación.
- Emplea una matemática compleja que evita que el clasificador realice sesgo hacia las muestras de entrenamiento.
- Versátil debido a que SVM es capaz de utilizar diferentes funciones Kernel para tomar la decisión final. La librería utilizada para lleva a cabo la clasificación tiene sus propios kernels. También se pueden crear nuevos.

Desventajas:

- Optimizado para clasificación biclase, aunque en sus diferentes extensiones permite la clasificación multiclase.
- Si el numero de características es mucho mayor que el numero de muestras no se obtienen buenos resultados.
- No proporciona directamente estimados de probabilidad, estos son calculados mediante el uso de validación cruzada.

Además, tenemos diferentes tipos de clases capaces de hacer clasificación multiclase, entre ellas se encuentran SVC, NuSVC y LinearSVC.

En la imagen se ve un ejemplo de cada una de ellas:

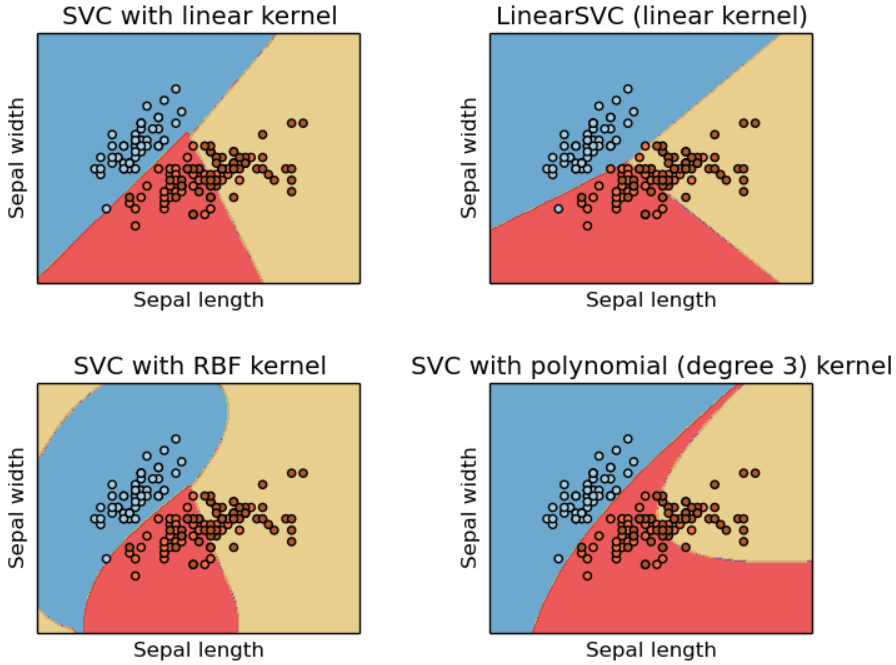


Figura 3.19: Proceso general sistema SVM.

### Entrenamiento Clasificadores

La fase de entrenamiento es muy importante para cualquier tipo de método de clasificación, como puede ser el que presentamos en este trabajo con base en la verificación del locutor. Esta fase consiste en crear un modelo compuesto por vectores de parámetros únicos de cada clase a clasificar que son los que contienen las características de la voz de

los locutores que disponemos en nuestra base de datos. Es por tanto necesario disponer de un conocimiento previo de las clases que se quiere clasificar para poder establecer los parámetros del clasificador.

Para las plantillas de características basadas en la voz, hemos editado dichas señales eliminando el ruido de fondo que presentan, aunque no por completo debido a que las muestras han sido tomadas con calidad inferior a la deseada, tratando de emular una situación real. Es necesario no tener solo muestras puras y perfectas debido a que después, a la hora de comprobar si un locutor es quien dice ser, no sabemos como será la prueba, puede que no sea una muestra impoluta. Sobre estos audios crearemos ficheros de características y que etiquetaremos en base a las clases sobre las cuales pretendemos llevar a cabo la clasificación. Es necesario llevar a cabo la etiquetación para que el clasificador sea capaz de modelar las fronteras de decisión y poder devolver el resultado esperado. Es entonces cuando, una vez etiquetado la totalidad de las muestras de las cuales disponemos para entrenar el sistema, crearemos un clasificador por cada locutor que sea capaz de diferenciarlo del resto.

**Test Clasificadores**

Esta fase sirve como referencia para determinar el tipo de clasificador o características a utilizar en el sistema basándose en el resultado que obtenido. En ella utilizaremos las muestras etiquetadas para evaluar el sistema y poder así determinar cuales son las mejores opciones dentro de nuestro problema de verificación de locutor.

Debido al carácter finito de nuestra muestra debemos ser capaces de maximizar su uso a la hora de entrenar y clasificar para no sesgar las características hacia una muestra que consideremos como válida. Por ello debemos implementar métodos de validación cruzada en la cual el resultado final que obtenemos en la clasificación no se encuentra sesgado a la muestra de clasificación.

## 3.4. Interacción con el usuario

El proceso a rasgos generales que seguirá el usuario a la hora de interactuar con el sistema biométrico lo vemos en la siguiente figura. En el momento en el que se le pida hablar, el sistema analizará su voz de forma que un software le comunique de si según las características utilizadas puede acceder al sistema o no. El desarrollo de la interacción se hará cuando al usuario se le pida una muestra de su voz. Este le enviará la información al

software que se encargará de procesarla y de mostrar al usuario la imagen o la aplicación correspondiente a lo que el sistema haya decidido.


Figura 3.20: Interacción usuario final.

# Capítulo 4

# Implementación

En este capítulo se explican cuales han sido los pasos seguido para la realización de este proyecto. En primer lugar se explica el despliegue del software necesario para el estudio. En segundo lugar se repasa el funcionamiento general de la parte de procesado de señal para acabar con la sección de clasificadores, la encargada de realizar la verificación del locutor.

## 4.1. Instalación Software

En este punto se explican brevemente las pautas para la instalación del software que se ha necesitado. Además en el Anexo podremos encontrar la programación desarrollada y la plataforma de Google Code con todo nuestro código subido, tanto de la parte correspondiente al procesado de señal como la parte de clasificadores.

### 4.1.1. Python

Python es el lenguaje utilizado en este proyecto, concretamente la versión 2.7[1]. Su instalación, para el sistema operativo Ubuntu, es tan sencilla como abrir la terminal y escribir el siguiente comando:

```
$ sudo apt-get install python2.7
```

[1]Página oficial de Python: http://www.python.org/

### 4.1.2. SciPy, Numpy y Matplotlib

Numpy es la librería principal que permite el uso de cálculo científico. Caracterizado por el uso de arreglos N-dimensionales. SciPy es una librería que utiliza los arreglos y manipulaciones posibles con NumPy y permite resolver problemas de integración , determinar máximos y mínimos (optimización), autovectores, procesamiento de imágenes, pruebas de estadísticas.

Para facilitar el uso de NumPy y SciPy[2] es recomendable instalar las librerías iPython y Matplotlib. iPython es una terminal que facilita el uso interactivo, mientras que Matplotlib permite visualizar gráficos 2D de calidad profesional.

La instalación de estos paquetes es tan sencilla como la instalación de Python, se permite instalar todos los paquetes a la vez con el siguiente comando, para el sistema operativo Ubuntu:

```
$ sudo apt-get install python-numpy python-scipy python-matplotlib ipython ipython
```

## 4.2. Funcionamiento General

El desarrollo de este sistema ha sido realizado con software libre, a excepción de la grabación y de la edición de audio, que han sido capturadas con ProTools y editadas con Adobe Audition 3.0. Sin embargo, el análisis de la señales, su procesado, clasificación y verificación de las voces ha sido implementado con software libre.

El estudio de Verificación Biométrica de Voz consiste en extraer características de la voz de diferentes individuos para crear plantillas y con nuevas muestras, verificar si un usuario es quien dice ser. Se trata de no hacer distinción de qué se dice, si no de quién lo dice.

Se parte de una base de datos compuesta por once locutores diferentes, cada uno aporta 10 locuciones diferentes, previamente editadas y sin demasiado ruido ni voces de fondo. La plantilla de cada usuario será una matriz de las características extraídas.

Para la extracción de patrones se utiliza MFCC (Mel Frequency Cepstral Coefficients) que sirve como técnica de extracción de características de la señal de voz. En la escala Mel, las frecuencias están posicionadas de forma logarítmica, de manera que es algo más cercano al oído humano. También se estudió la escala Cepstrum pero los MFCCs resultan

[2]Página oficial de Scipy y Numpy: www.scipy.org

más representativos, debido a que la escala Cepstrum está dividida en las bandas de frecuencia de forma linean obteniéndolas directamente de la FFT o de la DCT.

El proceso a seguir es:

1. Realizar la FFT de una señal previamente filtrada.
2. Crear el mapa logarítmico de amplitudes del espectro a partir de la escala de Mel, usando ventanas triangulares.
3. Hacer la DCT del mapa logarítmico de amplitudes, como si fuera una señal.
4. Los coeficientes obtenidos son las amplitudes del MFCC.

De esta forma se tiene como entrada una señal variable y su tasa de muestreo y como salida, las variables con la señal transformada.

En las siguientes secciones se explica cada uno de los pasos seguidos hasta llegar a la verificación del locutor.

### 4.2.1. Datos de entrada

Los audios han sido extraídos de una base de datos de voluntarios que ceden sus voces leyendo diferentes artículos. Se han cogido ocho locutores diferentes con diez locuciones por cada uno de ellos. Previamente se ha tenido que realizar una escucha de todas las locuciones con el objetivo de que todas tengan una amplitud suficiente, que el mensaje (aunque irrelevante) sea comprensible, un espectro claro y que no haya demasiado ruido de fondo ni voces externas que puedan confundir al sistema a la hora de clasificar a los locutores.Además se debía corroborar que cada uno de los locutores eran los mismos en cada una de las locuciones descargadas.

También se han realizado grabaciones en estudio de tres personas diferentes. Igual que en los casos anteriores, los locutores hablan y fueron grabados diez veces repitiendo diferentes mensajes con el objetivo de tener las mismas muestras para todo.

A excepción de las grabaciones realizadas en estudio, las descargadas tienen una calidad mediocre, ya que los medios con los que se capturaron y las condiciones no eran las ideales, pero es necesario tener muestras que no sean perfectas ya que a la hora de realizar el entrenamiento de los clasificadores, explicado en la sección 4.5.2, si solo se entrena con muestras ideales, a la hora de probar, si la muestra no es la ideal no será tan sencilla la

verificación.

### 4.2.2. Edición de audio

Debido a que no se van a implementar adaptaciones previas de la señal se ha tenido que editar las señales de forma que el ruido disminuya y los silencios iniciales de cada señal a analizar desaparezcan. De esta forma no se ha tenido que crear ningún algoritmo que reconozca en que momento empieza el locutor a hablar.

En la figura 4.1 se puede apreciar como la señal incluye ruido de una amplitud considerable como para que exista la opción confundir al sistema. Por consiguiente, en la figura 4.2 se aprecia la imagen sin ruido después de haberla filtrado y con los silencios iniciales eliminados.

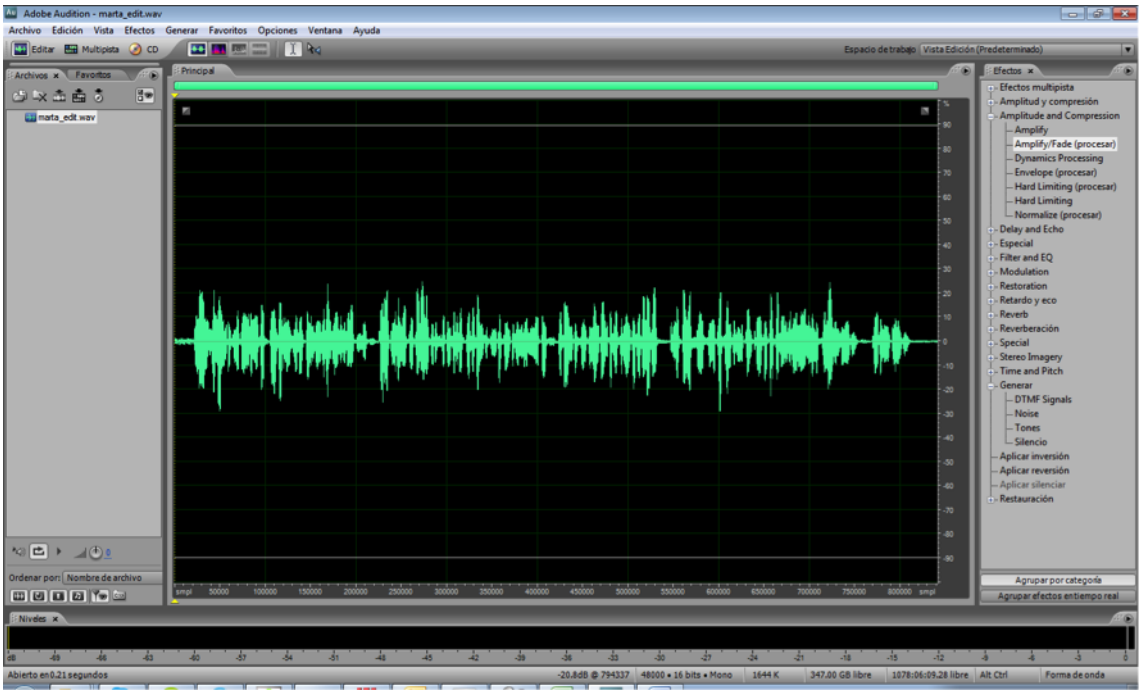


Figura 4.1: Señal de voz capturada sin tratamiento previo, con Adobe Audition 3.0.

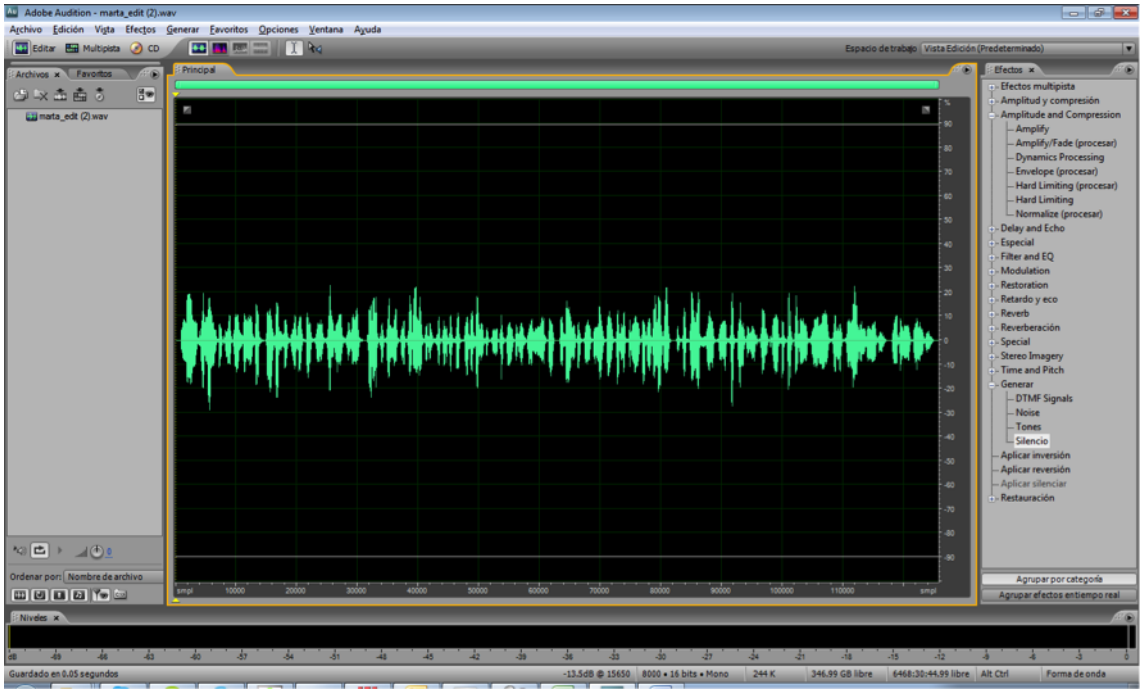


Figura 4.2: Señal de voz sin ruido ni silencios iniciales editada con Adobe Audition 3.0.

Estos ejemplos son de las grabaciones realizadas en estudio, y como se ve la amplitud es buena y la claridad de la señal también.

### 4.2.3. Filtrado Preénfasis

Consiste en un filtro paso alto (HPF) de primer orden [42]. Este filtro, en condiciones libres de ruido se suele utilizar a $\alpha = 0{,}95$. Esto hace que las zonas de altas frecuencias se enfaticen a la vez que las bajas frecuencias se desenfatizan. Es necesario utilizar un filtro preénfasis debido a que las características espectrales de la voz decaen con la frecuencia. Se debe compensar ese decaimiento para caracterizar correctamente la voz. En conclusión, se reduce el rango dinámico, ya que se debe tener en cuenta que la señal de voz suele tener una caída de 6 dB por cada octava.

En caso de que haya ruido en la señal de voz, este filtro no seria útil, ya que el ruido blanco es espectralmente plano. Y la señal de voz tiene la mayor parte de su energía en las bajas frecuencias, por lo que se estaría incrementando la zona del espectro predominada por el ruido.

El filtro utilizado en este caso ha sido el correspondiente a la siguiente ecuación:

$$x[n] = x[n] - \alpha x[n-1]; siendo \alpha = 0{,}95 \tag{4.1}$$

Aplicando este filtro preénfasis a cada uno de los segmentos en los que se ha dividido la señal, se compensa la atenuación producida en las altas frecuencias [6].

### 4.2.4. Division de la señal Entramado

La señal de voz varía a lo largo del tiempo, por lo que no se puede tratar directamente toda la señal al completo. Para ello se realiza una segmentación de la señal de voz. Se divide la señal en tramas de 30 ms con un desplazamiento de 15 ms. Obteniendo como resultado que dicho segmento de la señal de voz sea una señal cuasi-estacionaria.

El número de muestras por cada trama es de N y están separadas M muestras, creando un solapamiento de N-M. Este solapamiento se debe hacer para que exista una transición más suave entre una trama y la siguiente.

A la hora de la extracción de patrones, cada característica extraída se corresponde con una serie de coeficientes para cada uno de los segmentos. Esos coeficientes son con los que se debe caracterizar la señal de voz y con los que se tratará más adelante de verificar al locutor.

### 4.2.5. Enventanado

Cuando la señal se encuentra segmentada, se debe aplicar una ventana sobre la señal original [44].Enventanar una señal consiste en aplicar sobre la señal original en el tiempo una función limitada en tiempo, una ventana de N muestras.

$$x[m] = s[m]w[n-m]; m\epsilon[n-N+1, n] \tag{4.2}$$

La aplicación de esta depende del resultado a obtener. Hay que tener en cuenta la respuesta en frecuencia que va a tener la ventana ya que esto afecta al espectro.

En este estudio se trabaja en el dominio espectral, por lo que para la elección de la ventana se deben tener en cuenta el lóbulo principal y su anchura. Sin embargo, los lóbulos laterales solo se deben tener en cuenta los más altos, en caso de que afecten a la señal. Esto se hace, tal y como decíamos, para no afectar al espectro.

La programación del enventanado se realiza con las librerías Scipy [8]:

```
import numpy as np
from scipy.signal import get_window

M = 801 # tamano ventana
```

```
N = 1024 # tamano FFT
H = 400 # tamano salto

w = get_window(window, M)
```

En este estudio se han tenido en cuenta dos tipos de ventana, el resto se han descartado por la amplitud de los lóbulos laterales la ventana tipo Hamming y la ventana Blackman, las cuales se explican a continuación.

### Ventana Hamming

Esta ventana es un coseno elevado como la Hanning pero con unas pestañitas a los lados. Esto hace que tengamos una magnitud espectral con lóbulos laterales más amplios pero también más pequeños.

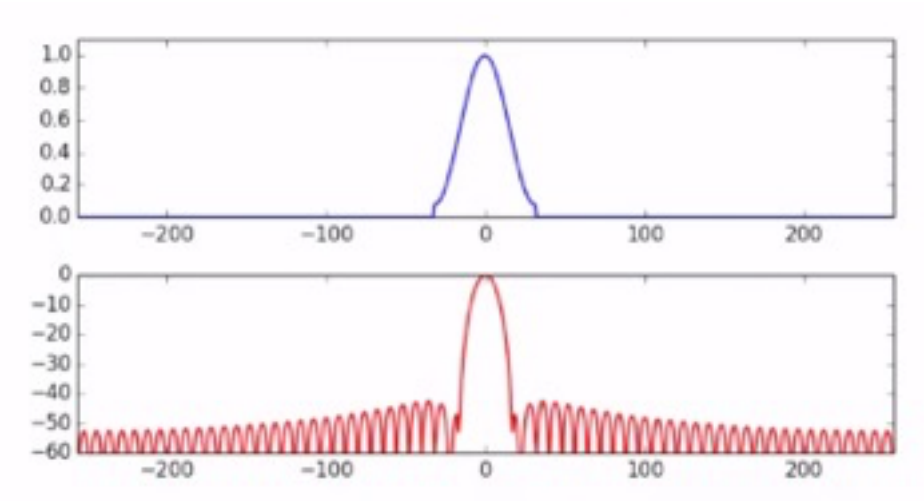


Figura 4.3: Forma de ventana Hamming

Lo ideal es tener los lóbulos laterales más pequeños posibles y el lóbulo principal muy ancho.

### Ventana Blackman

Es la suma de dos sinusoides, por lo que el lóbulo principal es más ancho y los lóbulos laterales son mas pequeños. Esta será la ventana utilizada debido al ancho del lóbulo y a que los lóbulos laterales no afectan demasiado al espectro.

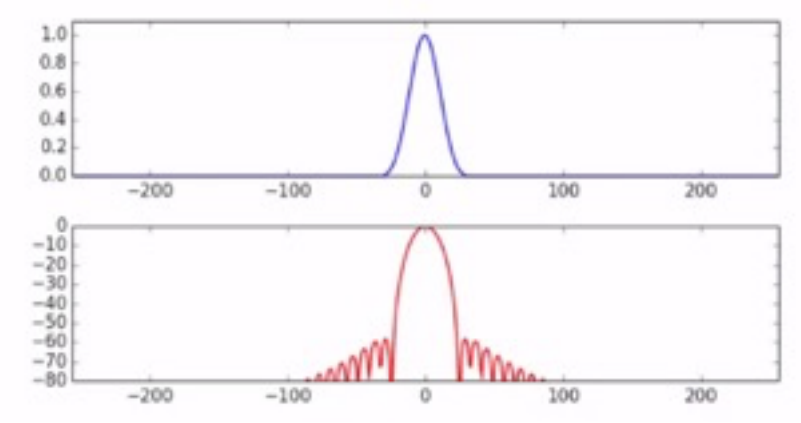

Figura 4.4: Forma de ventana Blackman

### 4.2.6. Tamaño de ventana

A la hora de elegir el tamaño de la ventana, se debe tener en cuenta cómo va a afectar a la magnitud y a la fase espectral. En la imagen 4.5 vemos dos ejemplos con un tamaño de ventana M = 128 y con otro tamaño de ventana para la misma señal de M = 1024.

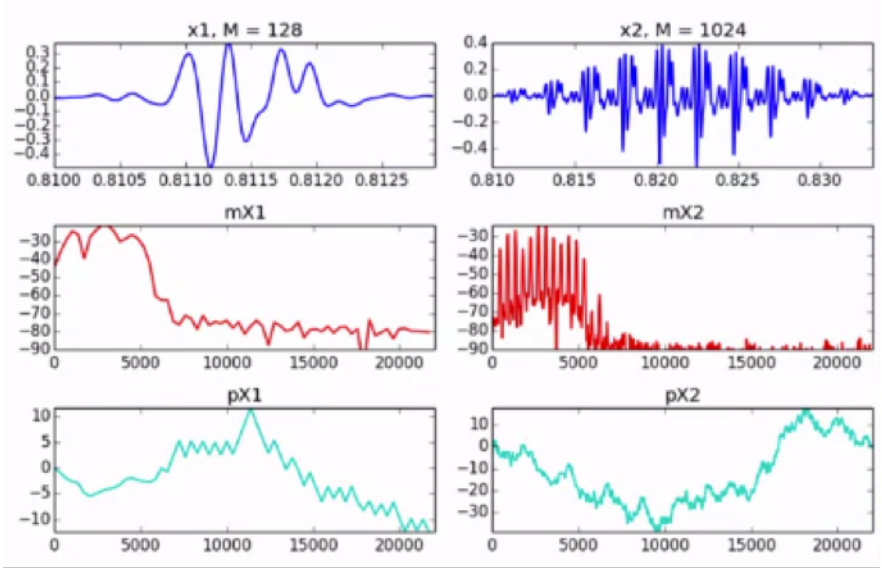


Figura 4.5: Diferentes tamaños de ventana para la misma señal.

La magnitud espectral de la ventana pequeña (mX1) tiene poca información, debido a que tiene pocas muestras (128) y en la ventana grande tenemos una señal mas detallada, nos da más información.

### 4.2.7. Ventana de tamaño par o impar

Esta característica no es tan importante como el tamaño, pero afecta. En la magnitud espectral son parecidas pero en la fase espectral cambia considerablemente. Ambas fases espectrales deben estar centradas en cero, pero en la ventana par, el resultado estará perfectamente centrado, debido a que para que esté centrado se debe tener una muestra en cero y por su condición de número par, tendrá diferentes números de muestras a cada uno de los lados. Por el contrario, en la ventana impar, tendremos perfectamente centrado el espectro y será simétrico.

Aunque la decisión de coger una ventana impar es fija, el número de muestras en la ventana se decide en función de la resolución que sea necesaria. En la imagen 4.6 se puede apreciar las diferencias de coger un tamaño de ventana u otro.

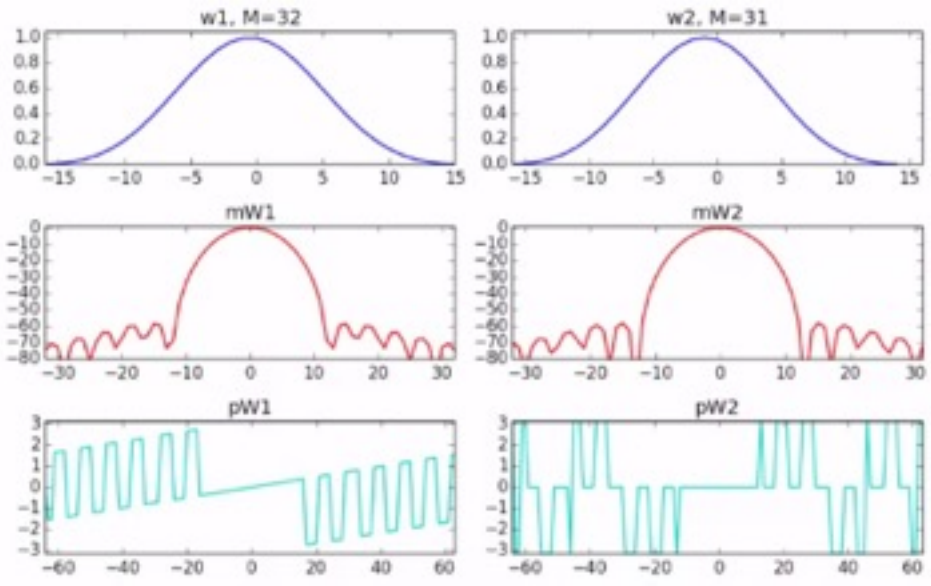


Figura 4.6: Tamaño ventana comparativa par o impar. Ejemplo implementado en Python y visualizado con Matplotlib.

### 4.2.8. Tamaño Fast Fourier Transform

La Fast Fourier Transform o la FFT es un algoritmo que permite obtener una representación de la señal de voz en el dominio de la frecuencia. La FFT es independiente del tamaño de la ventana, aunque debe ser mayor y siempre debe estar formada por un número de muestras igual a una potencia de dos. Es recomendable que sea grande ya que esto nos aporta mayor resolución en frecuencia y porque tenemos la posibilidad del zero padding [7].

Si por ejemplo se parte de una ventana de M = 512 y se coge un tamaño de FFT de N = 512 se obtiene como resultado unos cambios bruscos los cuales no dejan la información clara.

Si por el contrario se realiza una FFT mayor, como de 2048, haciendo un zero padding muy grande, se obtiene como resultado un espectro más suavizado el cual permite identificar las más cómodamente.

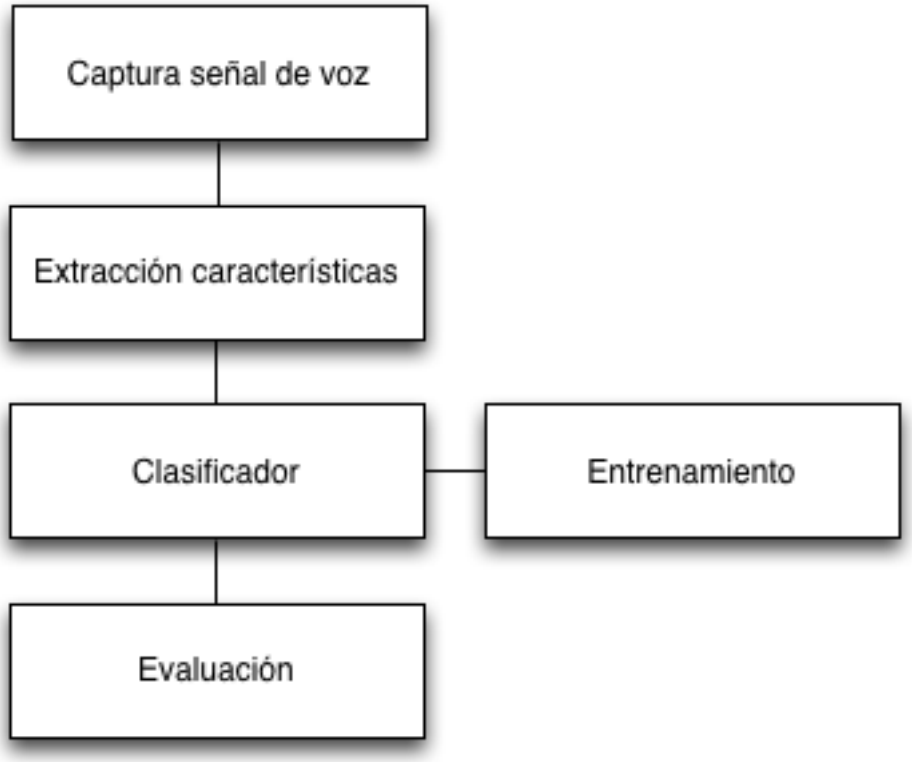


Figura 4.7: Tamaño FFT. (Ejemplo realizado con Python y librerías SMS-Tools).

### 4.2.9. Tamaño del salto

La ventana avanzará estas muestras después de la FFT. Por lo que es importante detenerse en elegir un tamaño de salto adecuado.

$$A_w[n] = \sum_{l=0}^{L} -1w[n - lH] = c \tag{4.3}$$

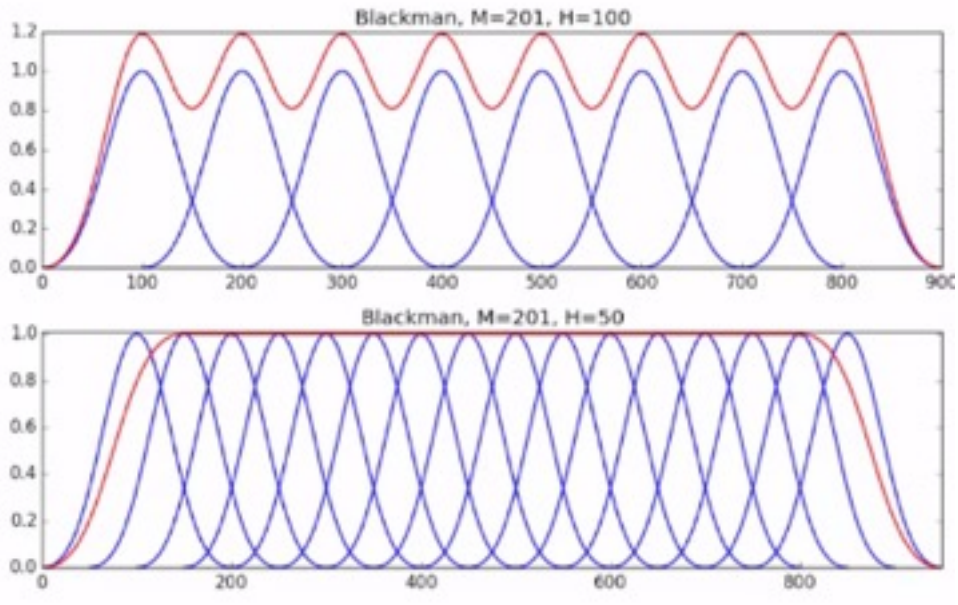


Figura 4.8: Tamaño salto ventana. (Ejemplo realizado con Python y librerías SMS-Tools).

En el primer ejemplo, en la figura 4.8 se aprecia una ventana de 201 muestras y saltamos cada 100, esto es, a la mitad. Obteniendo un resultado poco suavizado ya que tiene una oscilación, llamada modulación, y que se escuchará en la reconstrucción de la STFT.

En el segundo ejemplo, los saltos son de 50 muestras, y exceptuando en los lados que se puede apreciar una pequeña distorsión, el centro de la señal no se ve afectado de forma que se mantiene la identidad de la señal.

## 4.3. Short-Time Fourier Transform

Los sonidos reales no se pueden representar por un solo espectro. El sonido cambia en el tiempo y es necesario capturar esa variación. Para eso, la STFT es la solución.

Es una versión modificada de la DFT con dos importantes cambios: La señal de entrada no es solo x[n], si no que está multiplicada por la ventana de análisis. Esto es, x tiene un argumento que tiene las variables "n", "l."es el número de frame y "H."es el tamaño del salto. Se está produciendo un salto sobre "l", y "H"significa cómo de grande es dicho salto. Se deduce que "xçambia en tiempo dependiendo de "l H".

Como resultado, la salida no será un simple espectro, será una secuencia de espectros (x[k]). Siempre estará centrado en cero debido a que se ha elegido una ventana impar.

El enventanado es una manera de saltar y recorrer el sonido. Para comprender mejor el concepto se puede relacionar el espectro con una fotografía o imagen estática y la STFT

como un video.

## 4.4. Mel-Frequency Cepstral Coefficients

Después de realizar el procesado de la señal de voz para tenerla en el dominio frecuencial, se deben extraer las características. Para ello, la técnica elegida para este proyecto es la extracción de coeficientes MFCC (Mel-Frequency Cepstral Coefficients).

Los sonidos varían enormemente en el dominio de la amplitud e intentar identificar patrones en una señal acústica es una tarea compleja. Además, no representa la forma en como escucha un oído humano, que es en el dominio frecuencial.

Para solucionar esto, la escala Mel está basada en la variación conocida de los anchos de banda de las frecuencias críticas del oído. Propuesta por Stevens, Volkman y Newmann en 1937. El algoritmo desarrollado para la ejecución de estos coeficientes consiste en filtrar la señal mediante un banco de filtros formado por diferentes frecuencias y amplitudes con el objetivo de dar más resolución a las bajas frecuencias, tal y como hace el oído humano. Este filtrado se hace en el dominio frecuencial ya que previamente se ha realizado la STFT.

De la salida del filtro se calcula la energía en promedio y los valores obtenidos se ven como una nueva señal de tiempo discreto. Posteriormente se calcula la Transformada Inversa de Fourier y por último se obtienen dichos coeficientes.

Como resultado se consiguen entre 13 y 20 coeficientes diferentes, que son los conocidos como MFCC. En el apartado 3.2.4 se detalla cual es el significado de estos coeficientes.

Para el estudio realizado se ha optado por el uso de 13 coeficientes MFCC obtenidos de 26 bancos de filtros diferentes. La frecuencia mínima de la señal de entrada ha sido de 0 Hz y la máxima de la mitad de la frecuencia de muestreo. Para llevar a cabo el cálculo de la señal en el dominio frecuencial, el enventanado utilizado para la FFT ha sido de 512 muestras. Por último, las ventanas utilizadas para el análisis de características ha sido de 25 ms.

## 4.5. Clasificadores

Se analizan las descripciones del sonido desde sus niveles más bajo con el fin de describir sonidos de manera general. Se pueden agrupar características del audio en diferentes categorías. En este estudio se han dividido diferentes agrupaciones, categorizadas por el timbre de la señal (spectral centroid, MFCC), por su dinámica (volumen en un audio particular, nivel medio), por su pitch y, por último, categorizadas por patrones que describen señales en tiempo, esto es, aspectos del sonido que relacionan la evolución de un sonido con la textura del sonido. Esto lo podemos agrupar en características morfológicas (envelope of a sound, onset rate).

Partiendo de estos descriptores se puede hablar de colecciones de sonidos [44]. Descritas de la siguiente manera:

- Clustering de los sonidos.
- Clasificación de los sonidos.

Se fijan tres conceptos básicos: La similitud entre los sonidos, la agrupación de los sonidos y su clasificación. Una vez que se ha buscado la similutud entre los sonidos, estos se pueden agrupar teniendo en cuenta un criterio. Y finalmente, si se tienen clases marcadas, se puede proceder a clasificarlos, esto es, se pueden asignar clases a un sonido particular.

#### Clases de locutores

En este contexto, una colección o una clase de sonido se puede representar por un diagrama como el de la figura 4.9. Un sonido se define como un grupo de características. Cada característica tiene un valor numérico y para describir correctamente un sonido son necesarias varias características, pero para simplificar y entender el concepto, en el siguiente caso (Figura 4.9) solo se han tomado dos características.

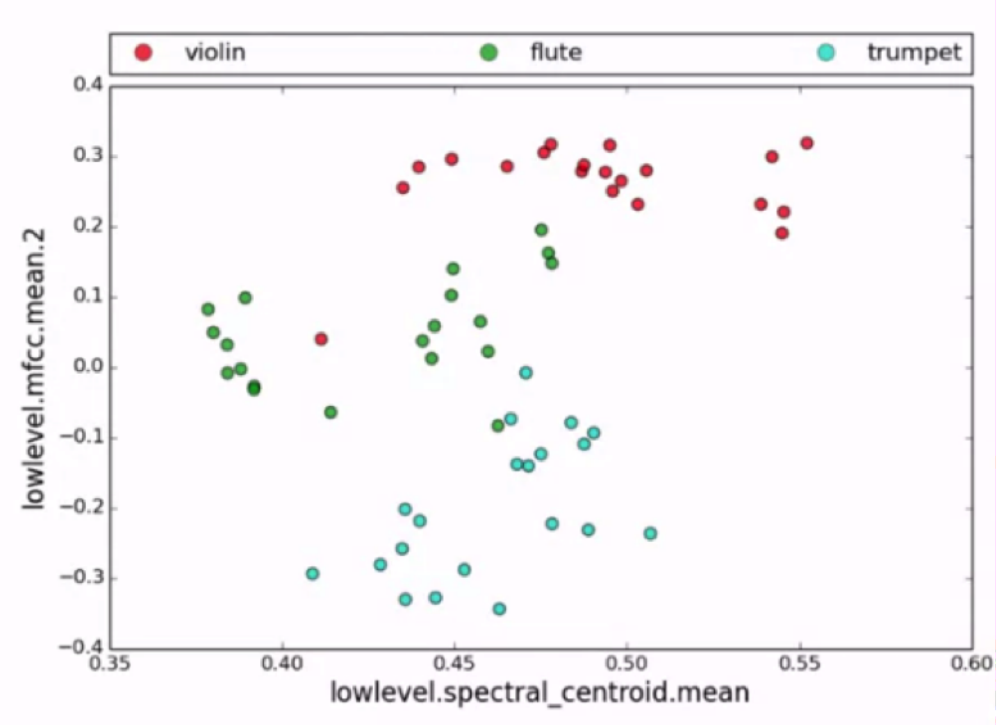


Figura 4.9: Colecciones de sonido. (Ejemplo realizado con Python y librerías SMS-Tools).

En el ejemplo mostrado en la figura 4.9 se han analizado las notas de 3 instrumentos; el violin, la flauta y la trompeta. El eje horizontal se corresponde a la media de los centroides espectrales extraidos y el vertical a la media de los coeficientes MFCC. Mediante este simple análisis se puede observar que los instrumentos musicales pueden ser agrupados en diferentes clases, estableciendo fronteras para la clasificación de nuevas muestras.

Si se considera que un sonido es representado por dos características, se puede visualizar en un espacio de dos dimensiones. Normalmente, en los problemas de clasificación de patrones, las características del objeto a clasificar suelen estar representadas por varias dimensiones, siendo su representación imposible de visualizar. Esto es debido a que la similitud entre las muestras para pocas características no permite establecer fronteras precisas para llevar a cabo el proceso de clasificación. Por ello es necesario operar en espacios dimensionales más altos. En el caso de este estudio, al utilizar 13 características, el espacio dimensional es demasiado grande para su visualización pero permite obtener buenas fronteras de decisión para diferentes tipos de locutores.

### 4.5.1. SVM

Tal y como se introdujo en el apartado de Reconocimiento dedicado a SVM, este tipo de clasificador trata de separar linearmente las clases maximizando la distancia entre las muestras de cada una [52]. Esta maximización de la distancia la realiza apoyándose en los vectores soporte que se encuentran en las fronteras entre las diferentes clases a separar. Un ejemplo de esta separación puede ser observada en la figura 4.10.

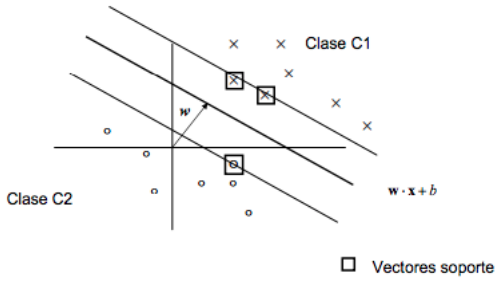


Figura 4.10: Ejemplo de SVM lineales.

En caso de que la dimensionalidad en la cual se encuentren las muestras de entrada no sea separable de manera directa, SVM aplica el llamado *Kernel trick* para elevar la dimensionalidad de las muestras de entrada y que puedan ser separables linearmente. En la figura 4.11 podemos observar como el clasificador SVM trata de generar un hiperplano que separe el espacio en dos o más regiones elevando los datos a una dimensión donde dicha separabilidad lineal pueda ocurrir [51].

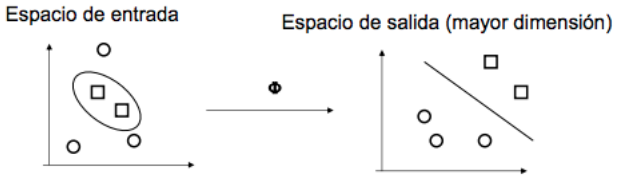


Figura 4.11: Ejemplo de SVM no lineales.

Aunque gracias a la matemática compleja que utiliza SVM, que es capaz de elevar las características hasta dimensiones infinitas, hay casos en los cuales dichos datos de entrada al clasificador no son separables linealmente. Es entonces cuando el kernel que se utiliza introduce fronteras de decisión no lineales en espacios dimensionales elevados, pudiendo realizar dicha separación de manera correcta.

### 4.5.2. Resultados

#### Entrenamiento y test

Para llevar a cabo el proceso de evaluación del sistema propuesto se dispone de 11 locutores, cada uno con 10 grabaciones con contenidos independientes. El cálculo de los patrones MFCC para cada locutor da como resultado una media de 5000 vectores de características para cada uno de ellos.

El entrenamiento se realiza con cantidades diferentes de muestras para poder diferenciar como afecta dicha cantidad al resultado final. Este entrenamiento se lleva a cabo con 200, 500, 1000 y 2000 muestras por locutor. El entrenamiento se realiza para cada locutor contra los restantes, en formato 1 contra N, obteniendo un clasificador por locutor.

La fase de test o evaluación se lleva a cabo con 10, 20, 50 y 100 muestras por cada locutor.

#### Análisis de Resultados

Una vez descritos en el capítulo 3 y el presente capítulo, los desarrollos experimentales que se han implementado en este estudio, se presentan los resultados obtenidos en la evaluación del rendimiento de las soluciones aportadas.

A continuación mostramos los resultados promediados para la combinación de muestras de entrenamiento con muestras de test. Los resultados se realizan para cada tipo de kernel elegido, lineal, rbf, polinomial y sigmoide, con diferentes valores del parámetro de penalización C.

Posteriormente a la visualización de los resultados, en la sección 4.5.3 se explican los resultados obtenidos.

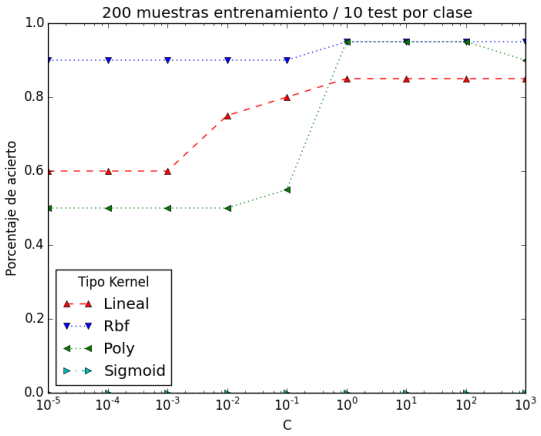


Figura 4.12: 200 muestras de entrenamiento y 10 de test.

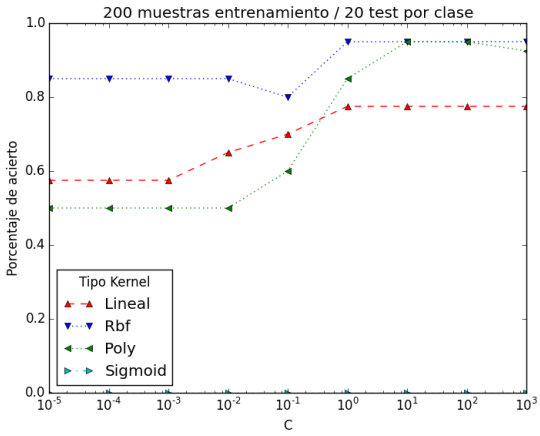


Figura 4.13: 200 muestras de entrenamiento y 20 de test.

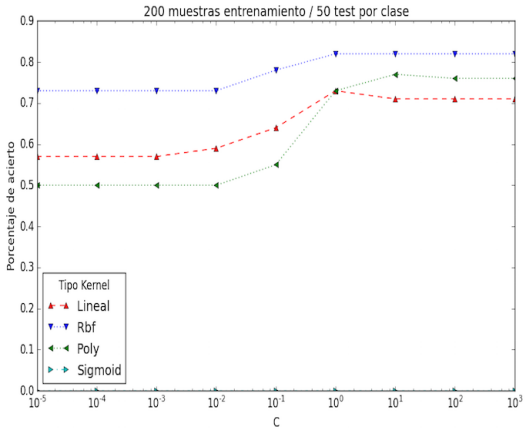


Figura 4.14: 200 muestras de entrenamiento y 50 de test.

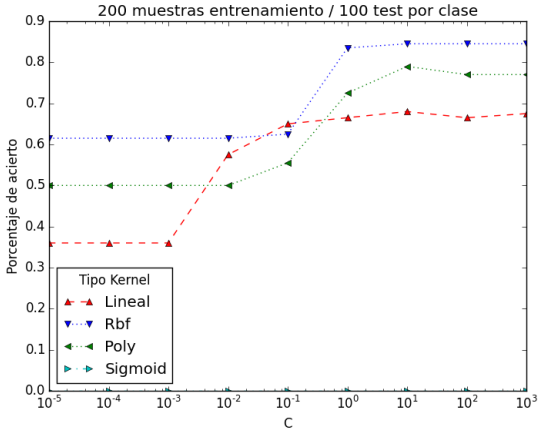


Figura 4.15: 200 muestras de entrenamiento y 100 de test.

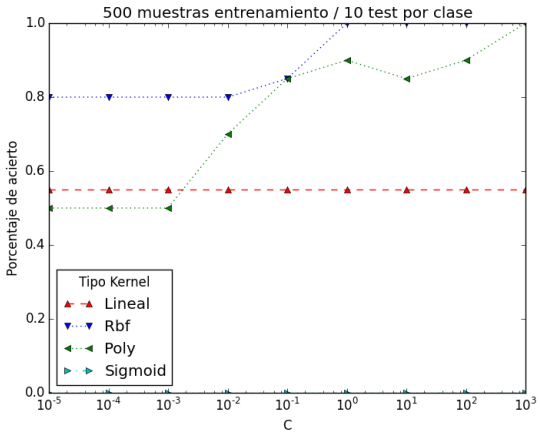


Figura 4.16: 500 muestras de entrenamiento y 10 de test.

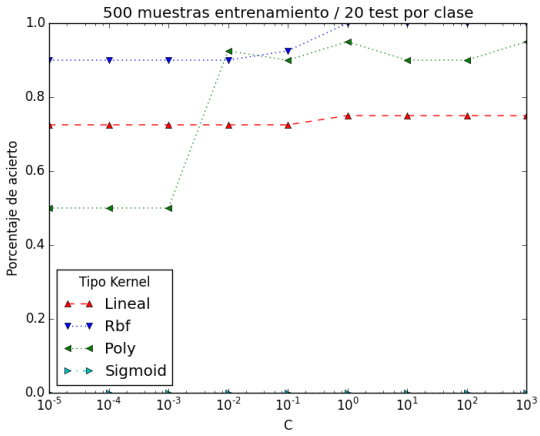


Figura 4.17: 500 muestras de entrenamiento y 20 de test.

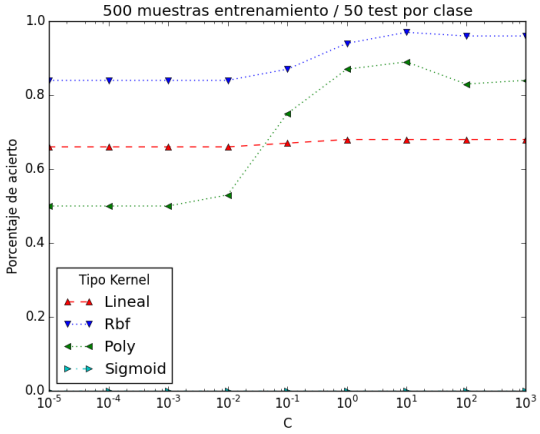


Figura 4.18: 500 muestras de entrenamiento y 50 de test.

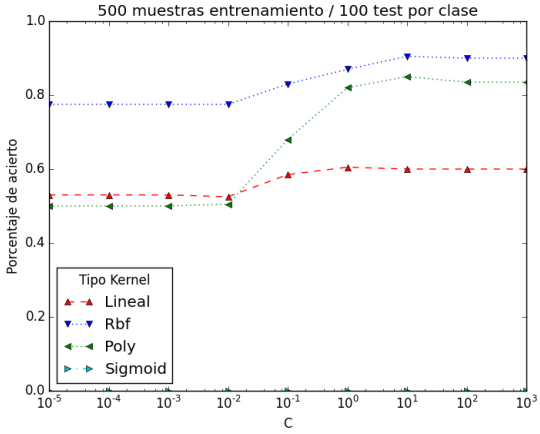


Figura 4.19: 500 muestras de entrenamiento y 100 de test.

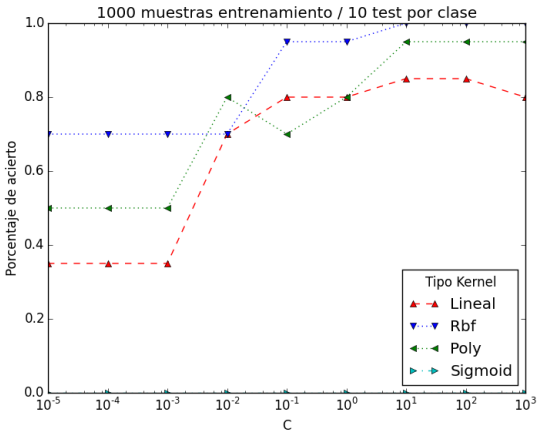


Figura 4.20: 1000 muestras de entrenamiento y 10 de test.

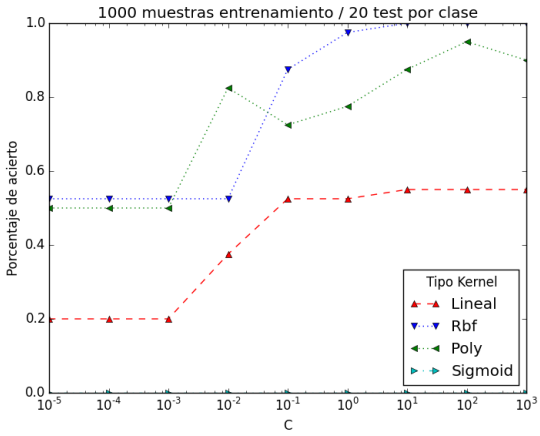


Figura 4.21: 1000 muestras de entrenamiento y 20 de test.

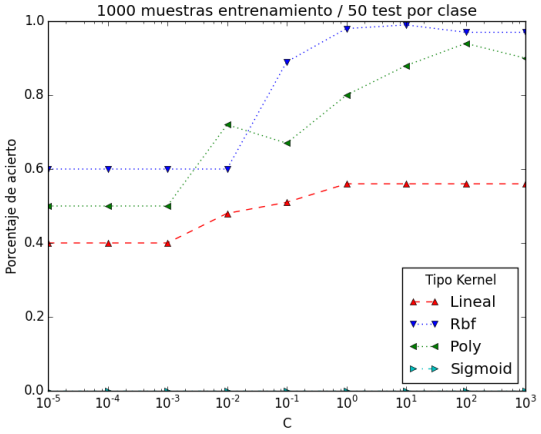


Figura 4.22: 1000 muestras de entrenamiento y 50 de test.

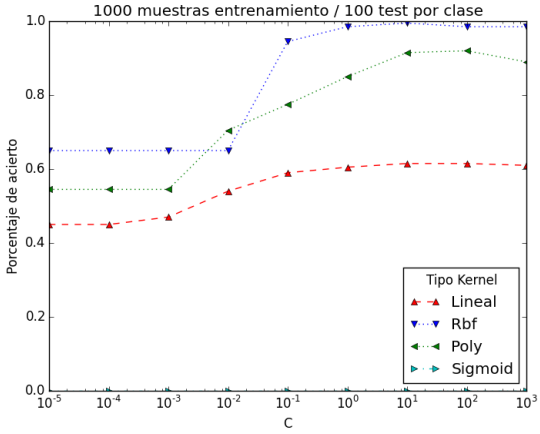


Figura 4.23: 1000 muestras de entrenamiento y 100 test.

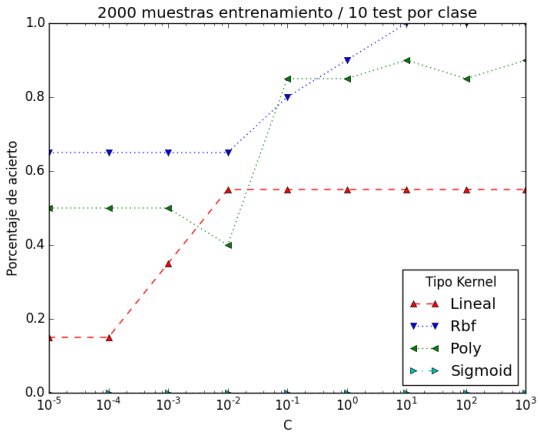


Figura 4.24: 2000 muestras de entrenamiento y 10 de test.

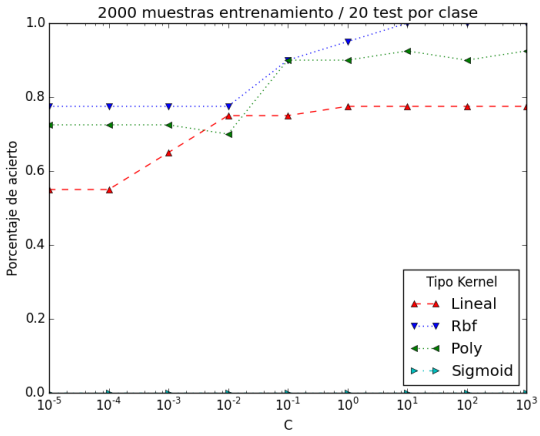


Figura 4.25: 2000 muestras de entrenamiento y 20 de test.

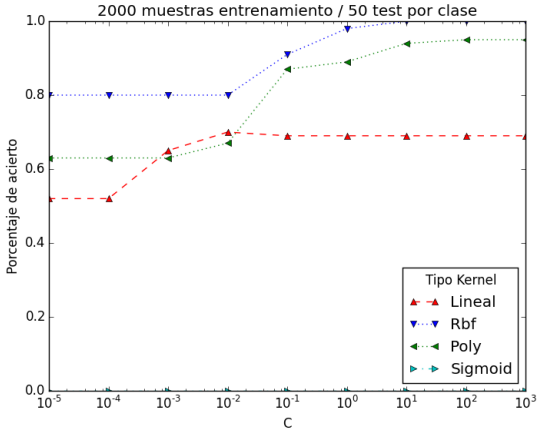


Figura 4.26: 2000 muestras de entrenamiento y 50 de test.

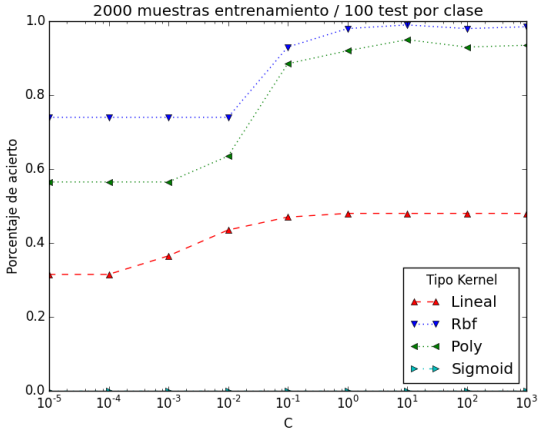


Figura 4.27: 2000 muestras de entrenamiento y 100 de test.

### Curva ROC

Como descripción global del sistema se presenta una curva ROC promedio con 2000 muestras de entrenamiento y 200 de test por cada locutor. El cálculo de esta curva ha sido optimizado para que elija los valores de C que mejor resultado aportan al sistema en cada tipo de Kernel.

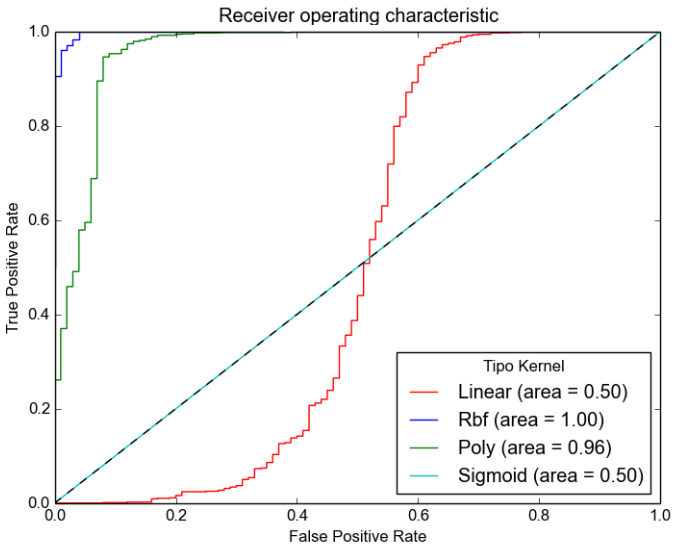


Figura 4.28: Curva ROC para 2000 muestras de entrenamiento y 200 de test.

### 4.5.3. Conclusiones

Los resultados obtenidos en el apartado anterior presentan diferentes tipos de conclusiones basadas en el método de entrenamiento y clasificación, los parámetros utilizados para SVM así como una conclusión global del método implementado.

En cuanto al método de entrenamiento y clasificación, al haber utilizado distintos valores de muestras de entrenamiento y de test, los resultados que más generalizan el rendimiento global del sistema son aquellos donde más muestra se utilizan. Este efecto es fácilmente observable cuando tenemos mayor cantidad de muestras de entrenamiento por clase así como de valores de test. Esto es debido a que si se dispone de poca cantidad de muestras, en el entrenamiento podemos tender a dar importancia a algunas características que no acaban de representar de manera adecuada al locutor que tratamos de hacer aprender al clasificador. Este mismo efecto puede producirse en el test y como consecuencia, no obtener los resultados deseados. Este concepto se denomina *sesgo* de la muestra.

Por otra parte, dependiendo de los parámetros que se aplican al clasificador,como el valor de C y el tipo de kernel, este queda entrenado de manera diferente y establece fronteras de decisión entre las clases de diferente manera. Cuanto mayor es el valor de C, menos error existe al establecer la frontera de decisión con los datos de entrenamiento. Aunque esto suponga un coste computacional elevado para llevar a cabo el entrenamiento, el resultado que se obtiene refleja que cuanto mayor es C mejor resultado se obtiene

en la clasificación. Hay que tener cuidado ya que puede llegar a darse también el caso de que si la C que se especifica es demasiado elevada puede llegar a sobreentrenar nuestro clasificador y no puede ser capaz de ser suficientemente generalista para otros datos que no sean los de entrenamiento. En cuanto al tipo de kernel, se puede afirmar que las características utilizadas para entrenar el clasificador no están lo suficientemente separadas linealmente, ya que con kernels no lineales la tasa de acierto es mucho mayor.

En lineas generales, partiendo de los resultados reflejados en la figura 4.28, el trabajo realizado es capaz de llevar a cabo el objetivo marcado de verificación. En esta última figura, los datos reflejados son aquellos que obtenemos entrenando de manera óptima los datos de entrenamiento para llevar a cabo la clasificación. Con un kernel $rbf$ se es capaz de obtener resultados de verificación óptimos para resolver el problema propuesto.

# Capítulo 5

# Conclusiones y trabajos futuros

En este capítulo final repasaremos el trabajo realizado en este Proyecto Fin de Máster. En primer lugar sintetizaremos las principales conclusiones obtenidas y presentaremos cuales han sido las principales aportaciones de nuestro estudio. A continuación se profundizará en los posibles trabajos futuros que pueden venir motivados como resultado de los trabajos realizados en este estudio.

## 5.1. Conclusiones

Hoy en día los sistemas de identificación de personas se han convertido en una necesidad para la sociedad. Los sistemas de identificación o verificación tradicionales (tarjetas o claves) se han ido sustituyendo por sistemas automáticos de reconocimiento biométrico. La utilización de la voz como rasgo biométrico aporta muchas ventajas frente a otros sistemas de identificación (universalidad, aceptación social, facilidad de uso, posibilidad de uso remoto, etc.).

En este Proyecto Fin de Máster se ha realizado el estudio y la implementación de un sistema biométrico de verificacion basado en la voz.

Compararemos los objetivos obtenidos con los planteados inicialmente con el fin de extraer conclusiones sobre el trabajo realizado. Para ello vamos a recordar los objetivos propuestos inicialmente:

- **Análisis de señales de voz.**

  Este objetivo ha sido cumplido con creces gracias a lo estudiado durante la carrera de Ingeniería de Telecomunicaciones. Además ha servido de refuerzo el curso realizado en www.coursera.org dirigido por Xavier Serra (Universidad

Pompeu Fabra) llamado Audio Signal Processing for Music Applications. Se ha aprendido a realizar tratamiento de señales acústicas, tanto en el dominio temporal como en el frecuencial, realizar síntesis de audio; modelos sinusoidales, armónicos, estocásticos; tracking de voz, transformaciones basadas en el dominio espectral y diferentes algoritmos de extracción de características entre otras cosas. Gracias a haber cumplido ampliamente este objetivo se ha podido trabajar con las señales de audio variando parámetros para poder adaptarlos a las necesidades que exigía este estudio.

- **Obtención de bases de datos de voz.**

  Una de las tareas para la realización de este Proyecto Fin de Máster era realizar grabaciones de diferentes locutores. Esto no fue necesario, aunque se hizo con algunas muestras, ya que se nos dio a conocer el proyecto VoxForge. Este proyecto tiene el objetivo de recoger transcripciones de textos mediante voz, para ser usados en Software de Reconocimiento de Voz.

- **Análisis y extracción de características humanas de la voz.**

  Este objetivo fue superado a la vez que el primero ya que, después de conocer el procesado básico de señales acústicas, se procedió al estudio de diferentes tipos de características, tanto en el ámbito musical como en el ámbito de la voz.

- **Comprobación de la existencia de signos distintivos únicos para cada una de las características obtenidas.**

  En una aproximación inicial, el estudio no se realizó con los coeficientes MFCC si no que se hizo con las formantes. El objetivo de utilizar esta característica en lugar de los MFCCs era tener una programación generalizada del sistema, esto es, que aunque cambiaran las características la programación siguiera siendo la misma. A pesar de que se suponía que con estas características el sistema no iba a ser capaz de verificar al locutor se realizaron las pruebas. El resultado fue el esperado, y el sistema no reconocía a prácticamente ninguno de los locutores, por lo que se procedió a la extracción de los coeficientes MFCC. En este caso se comprobó que sí existe un patrón para cada locutor, ya que el sistema, en su mayoría, verifica de qué locutor se trata.

- **Evaluar las posibilidades de la Biometría de Voz utilizando software libre.**

  El motivo principal de utilizar software libre es por los aportes de la comunidad, la rapidez de expansión y de solución de problemas y la libre implementación y despliegue de algoritmos. Como se ha comentado, se hizo una aproximación en Matlab, con el objetivo de ver si era un proyecto viable, pero el desarrollo íntegro se ha realizado en Python.

- **Analizar las posibilidades de verificación con sistemas sencillos de grabación (como micrófono de móvil o de ordenador)**

  Debido a que las locuciones se han cogido de usuarios aleatorios sin poder tener el control de cómo se realizaban las grabaciones ni bajo qué condiciones ambientales o técnicas, se puede dar este objetivo como cumplido.

- **Estudio de diferentes modelos de clasificadores para la verificación final.**

  Se han estudiado diferentes algoritmos tanto de clustering como de clasificadores con el fin de encontrar uno sencillo en su implementación y que nos aporte buenos resultados. Por eso hemos utilizado los clasificadores SVM modificando las variables de entrada, acotando el que mejor se ajustara y más preciso fuera a la hora de verificar si el locutor era quien decía ser.

- **Integrar las aplicaciones software realizadas en una única aplicación (PDS + SVM).**

  El objetivo final se ha cumplido satisfactoriamente, obteniendo buenos pero siempre mejorables resultados. Explicados en el siguiente apartado de trabajos futuros.

## 5.2. Líneas Futuras

En esta última sección, a partir de una crítica del trabajo realizado, se plantean mejoras y extensiones del mismo.

Por un lado, la carencia principal de este tipo de sistemas es que se puede saltar la seguridad de esta tecnología utilizando, por ejemplo, una grabadora de voz. Como propuesta para este intento de estafa se podría implementar una solución que aporte más características de forma que sea más restrictiva.

Debido a que se ha comentado que la voz cambia constantemente, se podrían proponer soluciones al usuario tales como grabar nuevas plantillas cada vez que se cambia de móvil, en caso de que fuera este a través del cual se verifica al locutor. De esta forma podríamos extraer un patrón de como cambia la voz y ver si existe alguna relación.

Como propuesta final, y debido a que este sistema pretende ser puesto en práctica con bases de datos de gran cantidad de locutores se podría plantear la reducción de dimensionalidad de las características para que el sistema pudiese trabajar en tiempo real.

# Capítulo 6

# Presupuesto

En los siguientes apartados veremos en tablas todos los costes que ha supuesto el proyecto. Ordenadamente explicaremos los costes en cuanto a programación esto es, software, en cuanto a hardware, en cuanto a trabajadores y en un último apartado veremos un resumen con el coste total que ha supuesto la realización del proyecto.

## 6.1. Costes Software

Los costes software se centran en las licencias necesarias para desarrollar el proyecto.

| Programa | Coste/Unidad (€) | Cantidad | Coste Total (€) |
|---|---|---|---|
| Sublime Text | 0 | x 1 | 0 |
| Python | 0 | x 1 | 0 |
| SciPy | 0 | x 1 | 0 |
| SVM | 0 | x 1 | 0 |
| Sketchbook Pro | 70 | x 1 | 70 |
| Adobe Audition 3.0 | 34,99 | x 1 | 34,99 |
| Pro Tools | 998,41 | x 1 | 998,41 |

Presupuesto Total Software: 1.103,4 €

## 6.2. Costes Hardware

En la próxima tabla están los costes de los elementos hardware para desarrollar el proyecto.

| Elemento | Coste/Unidad (€) | Cantidad | Coste Total (€) |
|---|---|---|---|
| MacBook Pro 13' | 1.529 | x 1 | 1.529 |
| Acer Aspire | 859 | x 1 | 859 |

Presupuesto Total Hardware: 2.388 €

## 6.3. Salario de los trabajadores

En la siguiente tabla se mostrará el salario base de cada trabajador dependiendo de su función.

| Trabajo a desarrollar | Trabajador | Precio/Hora (€) |
|---|---|---|
| Búsqueda de información | Ingeniero Junior | 20 |
| Configuración de equipos | Programador | 20 |
| Desarrollo Software | Programador | 20 |
| Pruebas finales | Ingeniero Senior | 30 |
| Gestión Proyecto | Ingeniero | 50 |

En la próxima tabla mostraremos los costes concretos:

| Trabajo a desarrollar | Nº de horas | Coste (€) |
|---|---|---|
| Búsqueda de información | 250 h | 5.000 |
| Configuración de equipos | 80 h | 1.600 |
| Gestión Proyecto | 400 h | 2.000 |
| Desarrollo Software | | |
| PDS | 250 h | 5.000 |
| SVM | 200 h | 4.000 |
| Pruebas finales | 80 h | 2.400 |

Presupuesto Total Trabajadores: 20.000 €

## 6.4. Coste Total

| Concepto | Coste (€) |
|---|---|
| Desarrollo Software | 1.103,4 |
| Desarrollo Hardware | 2.388 |
| Salario Trabajadores | 20.000 |

Presupuesto Total Proyecto: 23.491,4 €

# Bibliografía


[1] Marcos Ortega Hortas. (2013). Introducción a la Biometría. Técnicas Avanzadas de Procesado de Imagen.

[2] Ilsen About y Vincent Denis. (10-03-2011). Historia de la identificación de las personas. Ariel.

[3] Raymond B. Fosdick. (2013). Journal of Criminal Law and Criminology. Northwester University School of Law Scholarly Commons.

[4] María Mérida Aguilera. (03-2012). Reconocimiento Biométrico basado en imágenes de huellas palmares. Universidad Autónoma de Madrid.

[5] Pablo Pérez y Eduardo Álvarez y Susana de la Fuente y Laura García y Cristina Gutierrez. (12-2011). Estudio sobre las tecnologías biométricas aplicadas a la seguridad. Observatorio Inteco, 40-41.

[6] Ignacio Cobeta y Faustino Núñez y Secundino Fernández. (2013). Patología de la voz. Sociedad Española de Otorrinolaringología y Patología Cérvico-Facial, 1, 82-89.

[7] Fredrik Lindsten. (2010). A remark on zero-padding for increased frequency resolution. Sitio web: http://goo.gl/uBMFTw .

[8] SciPy Community. (2009). Tipos de ventanas y sus comandos en librerías SciPy. Sitio web: http://goo.gl/rGVxu .

[9] López-Cózar, R. y Callejas, Z. y Mctear, M.. (2006). Testing the performance of spoken dialogue systems by means of an artificially simulated user. Artificial Intelligence Review. Springer, Pág.: 291-323.

[10] Smithsonian Institution. (15-03-2008). Joseph Fabers Amazing Talking Machine of 1845. Sitio web: http://goo.gl/kaSMM7 .

[11] Ben Fino-Radin. (08-08-2012). When Machines Speak. Sitio web: http://goo.gl/783o4 .

[12] Iria Gallego G. e Irene Perez E. (2009). Sistema de reconocimiento de voz para los dígitos en inglés del 0 al 9. Universidad Carlos III de Madrid.

[13] Lars Kai Hansen. Programas en Matlab para la asignatura ?Non-linear signal processing?. Universidad Técnica de Dinamarca (DTU).

[14] Lars Kai Hansen. Non-linear signal processing, Hidden Markov Models. (2003). Anders Meng.

[15] Guner Arslan. (05-1998). Dual-Tone Multiple Frequency Detector Implementation. Sitio web: http://goo.gl/YOmqOJ .

[16] Carmen Sanchez. Aplicaciones de la Biometría a la Seguridad. Grupo de Biometría, Bioseñales y Seguridad Centro de Domótica Integral (CEDINT) Universidad Politécnica de Madrid. Sitio web: www.criptored.upm.es .

[17] Amadeo Guillemín. (1997). El mundo físico: el sonido. Montaner y Simón Editores. Pág: 250?279.

[18] Jorge Hierro Álvarez. (28-04-2004). Informe técnico sobre los sistemas de reconocimiento de voz. Departamente de prensa Assit.

[19] Andrzej Drygajlo. (2012) Automatic Speaker Recognition for Forensic Case Assessment and Interpretation. Springer. Pág: 21-22.

[20] Jose Valencia. (04-2014). Criptografía, Biometría y Esteganografía. Institución Universitaria Antonio Jose Camacho. Sitio web: http://goo.gl/zl5hE0 .

[21] Cristina Herrera y Begoña Morante. (04-2014). Aparato Fonador, Física y Fisiología. Universidad de Cantabria. Sitio web: http://goo.gl/zl5hE0 .

[22] Carlos Campaña. (03-2013). Aparato Fonador. Vox-Techonologies Sitio web: http://goo.gl/eJFWaH .

[23] Francesc Serratosa.(2010) La biometría para la identificación de las personas. Universitat Oberta de Catalunya.

[24] Bhanu, Bir y Chen, Hue (2008). Human ear recognition by computer. Springer.

[25] Chen, C. H. (2010). Handbook of pattern recognition and computer vision. Springer.

[26] Duda, Richard; Hart, Peter; Stork, David (2001). Pattern classification. Wiley.

[27] Escolano, Francisco y Suau, Pablo y Bonev, Boyán (2009). Information theory in computer vision and pattern recognition. Springer.

[28] Marcos Aular. (01-2013). Huellas Digitales - Computación Emergente. Universidad José Antonio Páez, San Diego.

[29] Dario D?Orazio, Simona De Cesaris, y Massimo Garai. (07-2011). A comparison of methods to compute the ?effective duration? of the autocorrelation function and an alternative proposal. Department of Energetic, Nuclear and Environmental Control Engineering (DIENCA),University of Bologna, Viale Risorgimento. Pág: 1954?1961.

[30] Chu, S. y Narayanan, S. y Jay Kuo C.(2012) Environmental sound recognition with time-frequency audio features. IEEE Trans. Audio, Speech and Lang. Pág: 1142-1158.

[31] Davis, S.y Mermelstein, P.(2001) Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoust., Speech, Signal Processing. Vol. 28(4), 1980, Pág.: 357?366.

[32] Wikipedia. (2008). Sitio web: https://es.wikipedia.org/wiki/Inarmon

[33] Fabien Gouyon. (2000). On the use of Zero-Crossing Rate for an Application of Classification of Percussive Sounds. Audiovisual Institute, Pompeu Fabra University,Barcelona. Sitio web: https://goo.gl/iZjBHh .

[34] J Salamon, E Gómez, J Bonada. (2011). Sinusoid Extraction and salience function design for predominant melody stimation.Music Technology Group Universitat Pompeu Fabra, Barcelona.

[35] P. M. Brossier. (2007). Automatic Annotation of Musical Audio for Interactive Application. QMUL, Londres. Sitio web: http://goo.gl/D9Jf11 .

[36] Antonio Aznar Casanova. De la Psicofísica Clásica a la Contemporánea. Facultad de Psicologia, Universidad de Barcelona. Sitio web: http://www.ub.edu/psicologiabasica/jaznare

[37] García, D., Aguilar, J., González, J. y Ortega, J. (2003). Support Vector Machine Fusion of Idiolectal and Acoustic Speaker Information in Spanish Conversational Speech. Proc. of the ICASSP?03. Pág: 229-232.

[38] Gómez, P., Martínez, R., Díaz, F., Lázaro, C., Álvarez, A., Rodellar, V., Nieto, V. . (04-2005) Estimation of vocal cord biomechanical parameters by non-linear inverse filtering of voice. Int. Conf. on Non-Linear Speech Processing, Barcelona.

[39] Cristian Luna, Ignacio Bevacqua y Nicolás Salvay. (05-2011). Análisis del sistema de fonación humano. Universidad Tecnológica Nacional, Facultad Regional de Córdoba.

[40] Oxana Pérez, Fernando Poceros y Alexis Villalobos. (2013). Sistema de Seguridad por Reconocimiento de Voz. Escuela Superior de Ingeniería Mecánica y Eléctrica.

[41] Dra. Begoña Torres. Anatomía Funcional de la Voz.

[42] Xuedong Huang. (2001). Spoken Language Processing: A Guide to Theory, Algorithm and System Development.

[43] Lawrence Rabiner y Ronald Schafer. (2010). Theory and Applications of Digital Speech Processing.

[44] Xavier Serra. Audio Signal Processing for Music Applications. Universidad de Stanford. Curso impartido a traves de www.coursera.org.

[45] B. Gold, N. Morgan, D. Ellis. (2011) ?Speech and Audio Signal Processing ? Processing and Perception of Speech and Music?, Wiley, 2nd edition.

[46] J. Benesty, M. M. Sondhi, Y. Huang (eds.). (2008). Handbook of Speech Processing, Springer.

[47] J. Deller, Jr., J. H. L. Hansen, Springer ?Handbook of Speech Processing and Speech Communication?, Springer, 2008.

[48] L. R. Rabiner and W. Schafer. (2009). Theory and Application of Digital Speech Processing?, Prentice Hall Inc.

[49] E. Alpaydin. (2010). Introduction to Machine Learning, The MIT Press, 2nd edition.

[50] Richard O. Duda, Peter E. Hart, David G. Stork. (2000). Pattern Classification, 2nd Edition.

[51] A. Stolcke. (2008). Machine Learning for Speaker Recognition. NIPS. Workshop on Speech and Language: Learning-based Methods and Systems.

[52] W. M. Campbell, D. E. Sturim, D. A. Reynolds. (2006). Support vector machines using GMM supervectors for speaker verification. IEEE Signal Proc. Letters 13(5), 308-311, 2006